

Sprawozdanie 1

Modelowanie i Identyfikacja

Maciej Kajdak
Nr indeksu: 226256

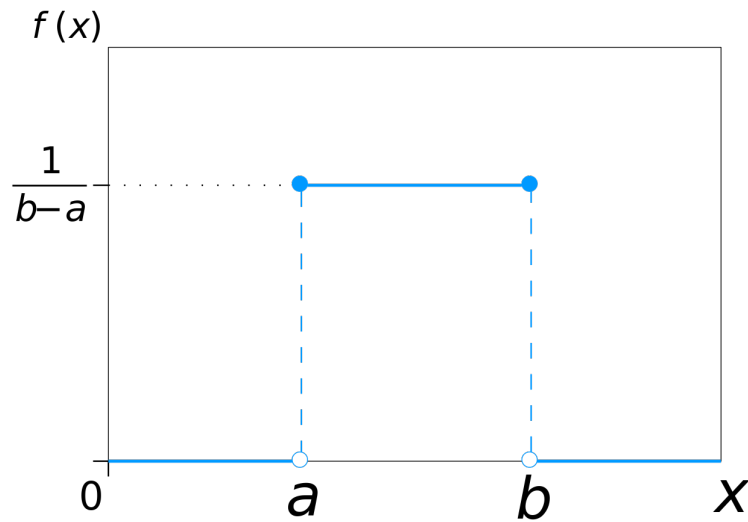
28 marca 2019

1 Wstęp

Przy generowaniu liczb losowych w najprostszym przypadku można posłużyć się tzw. generatorami fizycznymi. Do nich można zaliczyć chociażby rzut monetą (losowanie liczb ze zbioru $0, 1$) lub pobieranie co pewien kwant czasu próbki z przebiegu pewnego szumu elementu elektronicznego. Takie rozwiązania są jednak pod wieloma względami niepraktyczne. Postęp, któremu z biegiem lat ulega nauka i technologia pozwala na coraz coraz wydajniejsze programowe generowanie liczb losowych. Do celu generowania liczb można wykorzystać np. generatory liczb losowych o rozkładzie równomiernym. Wśród nich najczęściej spotykane są generatory liniowe, nieliniowe, Fibonacciego lub generatory oparte na rejestrach przesuwnych. Chcąc wygenerować liczby losowe o dowolnym rozkładzie prawdopodobieństwa również można posłużyć się algorytmami powszechnie znanymi, takimi jak np. metoda odwracania dystrybucyjności albo metoda eliminacji. Ważnym aspektem generacji liczb losowych jest także testowanie generatorów. Dzięki temu możliwe jest ocenianie jakości generatora pod konkretnym kątem. W niniejszym sprawozdaniu opisane zostały wyniki badań, którym zostały poddane różne algorytmy generowania liczb losowych. Generatory implementowano w środowisku Python w wersji 3.7.1 [3] wraz z pakietami numpy w wersji 1.15.4 [2] oraz matplotlib w wersji 3.0.2 [1]. Wszystkie stworzone w ramach laboratoriów skrypty można znaleźć w [repozytorium twórcy w serwisie Github](#)

2 Generator liczb losowych o rozkładzie jednostajnym

Jednostajny (równomierny) rozkład prawdopodobieństwa na przedziale $[a, b]$ charakteryzuje się tym, że jego gęstość na tym przedziale jest stała i różna od zera, a poza wspomnianym przedziałem przyjmuje wartości 0 jak zostało to przedstawione na rysunku 1.

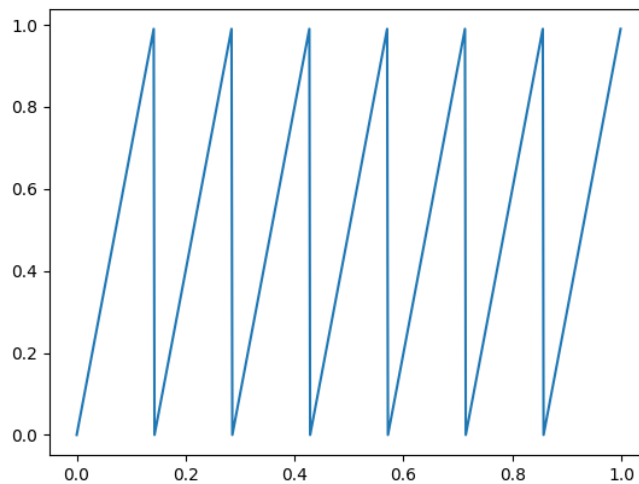


Rysunek 1: Gęstość rozkładu jednostajnego.

Zaimplementowano i przetestowano 2 rodzaje generatorów: oparty na przebiegu piłokształtnym oraz generator liniowy. Wyniki badań zostały przedstawione w dalszej części sprawozdania.

2.1 Generator oparty na przekształceniu piłokształtnym

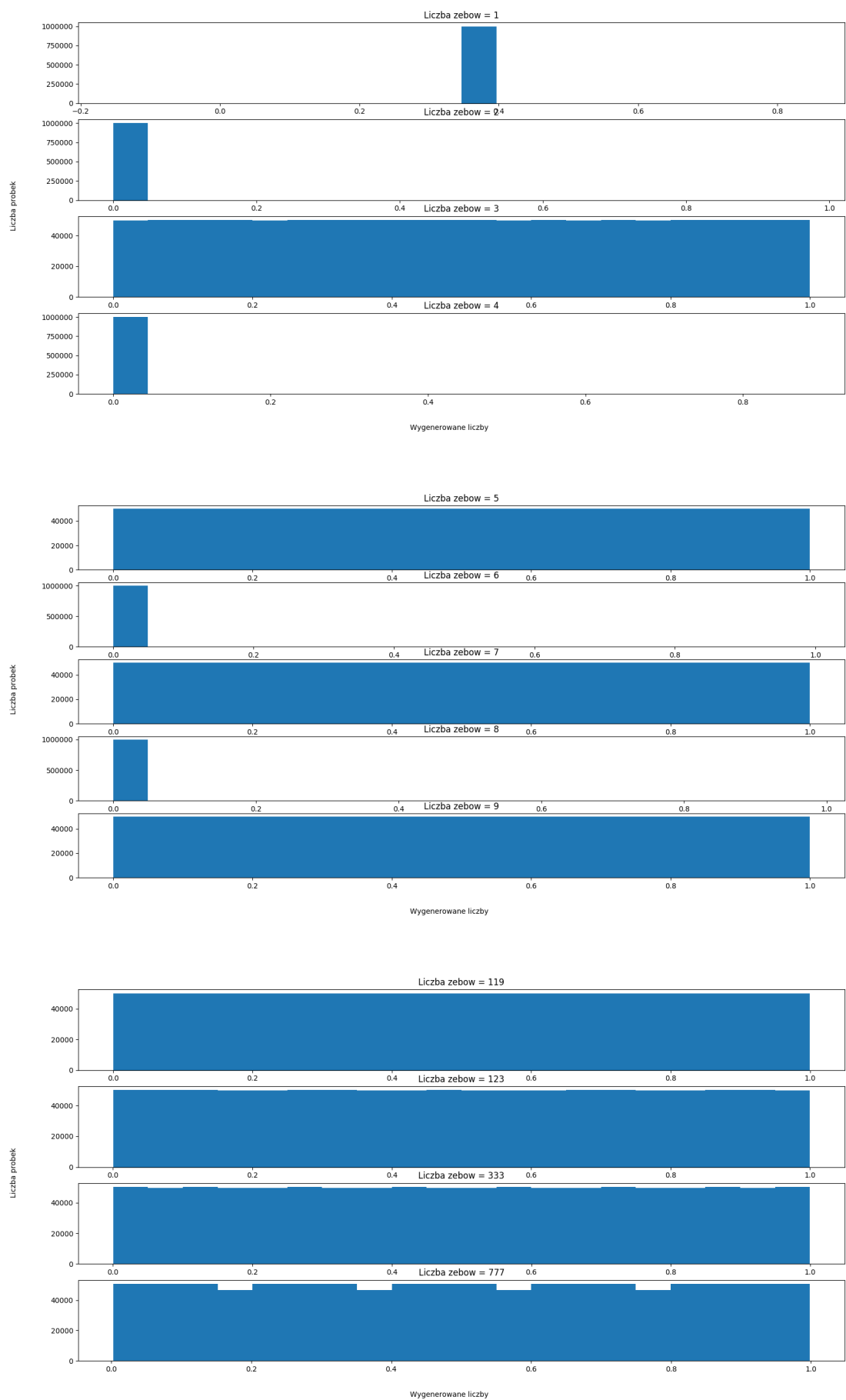
Sygnal piłokształtny zawdzięcza swoją nazwę graficznej reprezentacji, która przypomina zęby piły (rysunek 2). Odpowiednie próbkowanie takiego sygnału pozwala na generowanie liczb losowych o rozkładzie jednostajnym.



Rysunek 2: Przebieg piłokształtny o 7 „zębach” wygenerowany na przedziale $[0, 1)$ w środowisku Python

Generacja liczb losowych z takiego przebiegu opiera się o algorytm, którego istotą jest równanie 1.

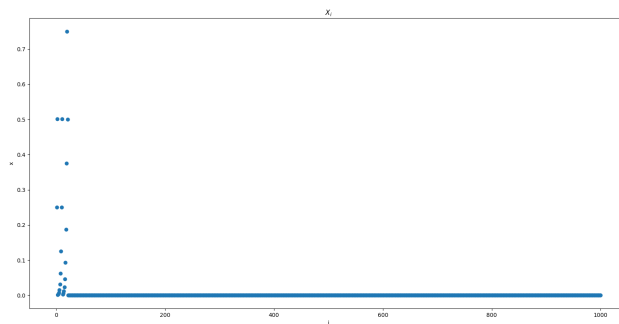
$$X_{n+1} = X_n \cdot Z - \lfloor X_n \cdot Z \rfloor \quad (1)$$



Rysunek 3: Histogramy generowanych próbek z przebiegu piłokształtnego

Jak można od razu zauważyć dla liczby zębów piły, czyli $Z = 1$ histogram przedstawia 1 słupek. Dzieje się tak, gdyż równanie 1 przyjmuje w takim wypadku postać 2 co oznacza, że każdy kolejny element jest powtórzeniem poprzedniego. Dodatkowo dla parzystej liczby zębów również występuje problem skupienia zdecydowanej większości liczb w jednym przedziale. Sprawdzając poszczególne liczby wygenerowane dla takiej ilości „zębów” (rysunek 4) można zauważyć, że liczby generowane są na całym przedziale, lecz po pewnym czasie wszystkie liczby są zerami. Dzieje się tak dlatego, że podczas mnożenia przez 2 w pewnym momencie „zerują się” dalsze miejsca po przecinku co prowadzi do sytuacji, gdzie każda kolejna liczba będzie zerem.

$$X_{n+1} = X_n - \lfloor X_n \rfloor \quad (2)$$



Rysunek 4: Wykres kolejnych generowanych liczb dla liczby „zębów” piły równej 2

2.2 Generator liniowy

Badając o właściwości generatorów liniowych ma się na myśli generatory zapisywane w postaci przedstawionej jako równanie 3

$$X_{n+1} = (a_1 \cdot X_n + a_2 \cdot X_{n-1} + a_3 \cdot X_{n-2} + \dots + a_k \cdot X_{n-k+1} + c) \bmod m \quad (3)$$

, gdzie $a_1 - a_k$, c , m są pewnymi stałymi parametrami generatora, a operacja $\bmod m$ oznacza wyznaczanie reszty z dzielenia przez m .

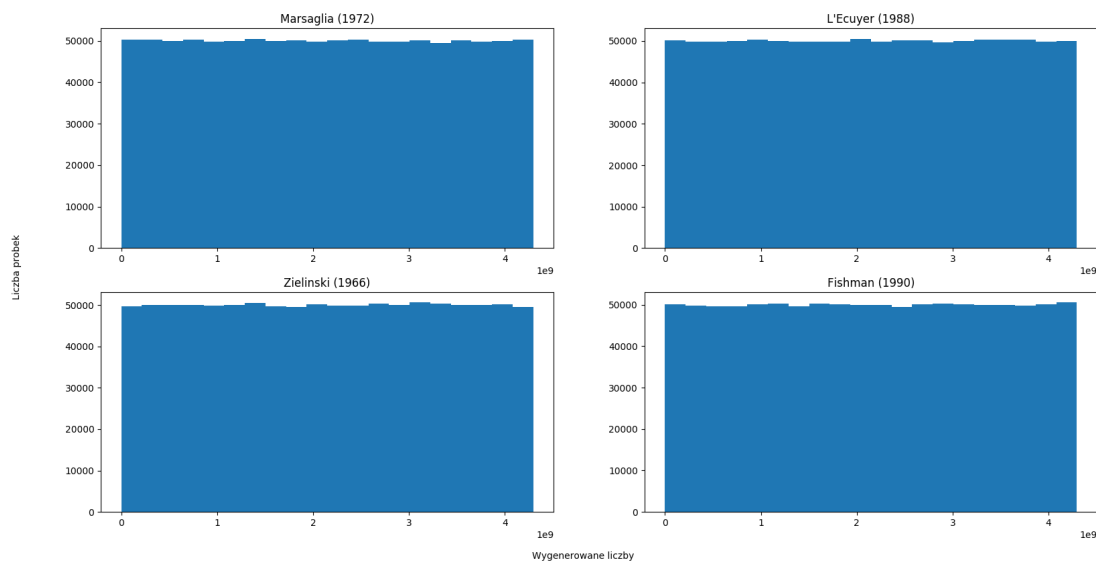
W szczególnym przypadku, gdy $k = 0$ to opis generatora przyjmuje postać równania 4. Co więcej jeżeli $c = 0$ to taki generator nazywany jest generatorem multiplikatywnym, z kolei, gdy $c \neq 0$ – generatorem mieszanym.

$$X_{n+1} = (a \cdot X_n + c) \bmod m \quad (4)$$

Do celów badań zaimplementowano 3 generatory multiplikatywne oraz generator mieszany, a ich parametry przedstawiono w tabeli 1. Zestawy parametrów zaczerpnięto z książki Wieczorkowskiego i Zielińskiego [7]. Wygenerowano przy jego pomocy po 100000 liczb dla każdego zestawu parametrów. Wyniki przedstawiono w postaci histogramu dzielącego przedział $[0, m)$ na 10 podprzedziałów. Histogram został przedstawiony na rysunku 5. Jak widać histogramy przypominają (z niewielkimi odchyleniami) wykres gęstości rozkładu jednostajnego.

a	c	m	Źródło
69069	1	2^{32}	Marsaglia (1972)
40692	0	$2^{31} - 249$	L'Ecuyer (1988)
$2^2 \cdot 23^7 + 1$	0	2^{35}	Zielinski (1966)
1099087573	0	2^{32}	Fishman (1990)

Tablica 1: Tabela parametrów generatorów liniowych.



Rysunek 5: Histogramy wyników generacji liczb losowych generatora liniowego dla poszczególnych zestawów parametrów.

3 Metoda odwrotnej dystrybucyjności

Chcąc wygenerować liczby losowe o dowolnym rozkładzie można wykorzystać algorytm nazywany metodą odwrotnej dystrybucyjności. Jak sama nazwa wskazuje, bazuje ona na wykorzystaniu dystrybucyjności rozkładu do generowania liczb. Podstawowym jednak założeniem jest, że „jeżeli istnieje gęstość g rozkładu μ na \mathbf{R} , to $F_\mu(t) = \int_{-\infty}^t g(x)dx$ ” [4]. Dodatkowo niech U będzie zmienną losową o rozkładzie jednostajnym na $[0, 1)$. Dzięki takim założeniom można wyznaczyć nową zmienną losową $X = F^{-1}(U)$, wtedy zmienna losowa X ma rozkład prawdopodobieństwa o dystrybucyjności F . Zatem generując ciąg liczb $X_i = F^{-1}(U_i)$ otrzymuje się ciąg liczb losowych o rozkładzie z dystrybucyjnością F .

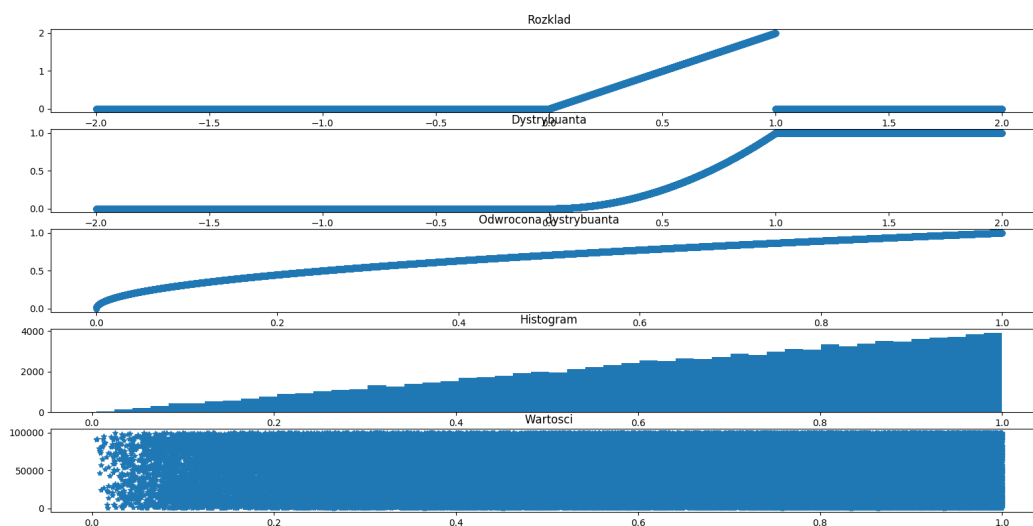
W taki sposób zaimplementowano 4 generatory liczb losowych o rozkładach:

- dowolny,
- trójkątnym,
- wykładniczym,
- Laplace’a.

3.1 Rozkład dowolny

Dla takiego rozkładu gęstość (przedstawiona wcześniej w postaci wykresu ogólnego na rysunku 3) można przedstawić jako

$$f(x) = \begin{cases} 2x, & \text{dla } x \in [0, 1] \\ 0, & \text{dla } x \in (-\infty, 0) \cup (1, \infty) \end{cases} \quad (5)$$

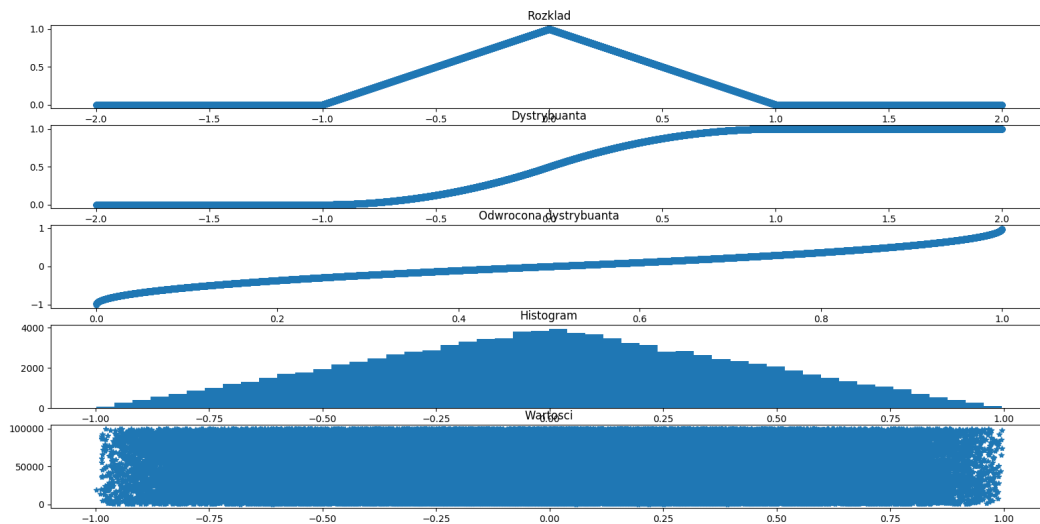


Rysunek 6: Gęstość rozkładu, wykres dystrybucyjny oraz jej odwrotności, histogram wygenerowanych liczb oraz wygenerowane liczby z rozkładu równomiernego

3.2 Rozkład trójkątny

Gęstość rozkładu prawdopodobieństwa dla implementowanego generatora wynosi:

$$f(x) = \begin{cases} x + 1, & \text{dla } x \in (-1, 0) \\ x - 1, & \text{dla } x \in [0, 1) \\ 0, & \text{dla } x \notin (-1, 1) \end{cases} \quad (6)$$

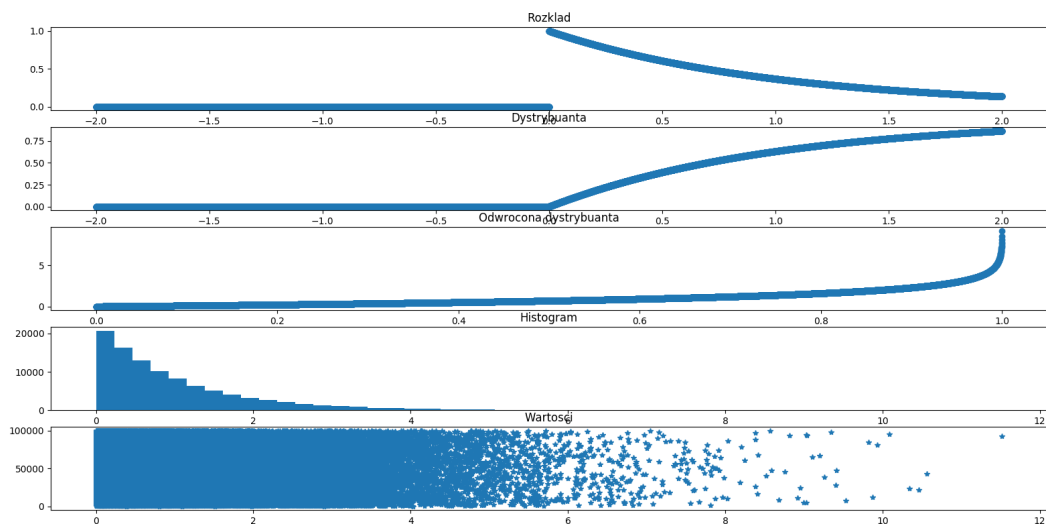


Rysunek 7: Gęstość rozkładu, wykres dystrybuanty oraz jej odwrotności, histogram wygenerowanych liczb oraz wygenerowane liczby z rozkładu trójkątnego

3.3 Rozkład wykładniczy

Gęstość wykładniczego rozkładu prawdopodobieństwa:

$$f(x) = e^{-x}, \quad x \geq 0 \quad (7)$$

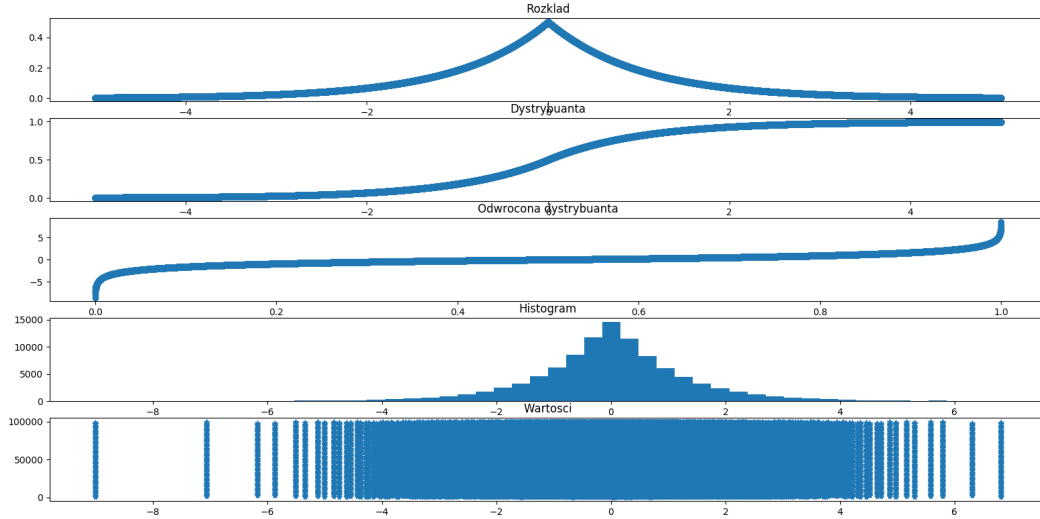


Rysunek 8: Gęstość rozkładu, wykres dystrybuanty oraz jej odwrotności, histogram wygenerowanych liczb oraz wygenerowane liczby z rozkładu wykładniczego

3.4 Rozkład Laplace’a

Gęstość rozkładu Laplace’a w ogólnym przypadku przedstawia się jako:

$$f(x) = \frac{1}{2b} \begin{cases} e^{-\frac{\mu-x}{b}}, & \text{dla } x < \mu \\ e^{-\frac{x-\mu}{b}}, & \text{dla } x \geq \mu \end{cases} \quad (8)$$



Rysunek 9: Gęstość rozkładu, wykres dystrybuanty oraz jej odwrotności, histogram wygenerowanych liczb oraz wygenerowane liczby z rozkładu Laplace’a

4 Metoda odrzucania

Algorytm generacji liczb losowych w metodzie odrzucania wiąże się z wykorzystaniem dwóch gęstości rozkładów: $f(x)$ i $g(x)$, czyli kolejno gęstości rozkładu prawdopodobieństwa zmiennej X , której realizacja ma zostać wygenerowana oraz pomocnicza gęstość innej zmiennej losowej. Dodatkowo wymagane jest istnienie parametru $c > 0$ dla którego spełniona jest zależność 9.

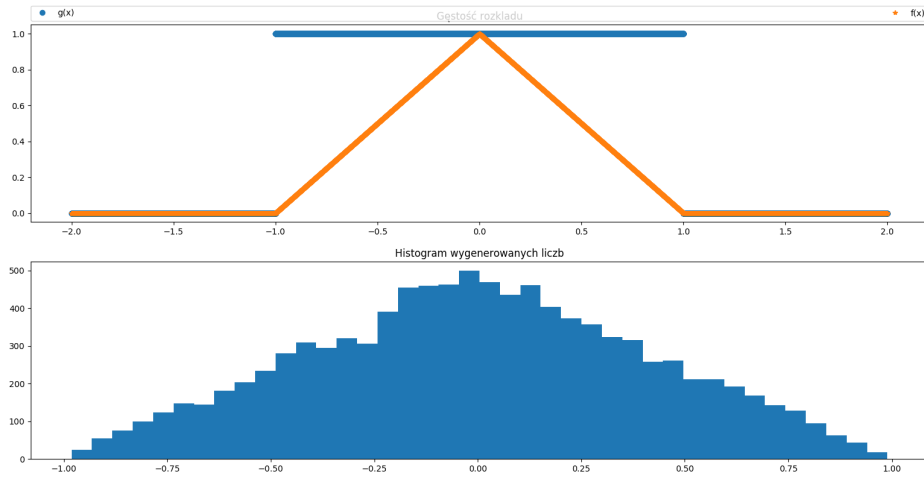
$$f(x) \leq cg(x), \text{ dla } x \in \mathbf{R} \quad (9)$$

4.1 Rozkład trójkątny

Dla przykładu przyjęto funkcję gęstości rozkładu prawdopodobieństwa z równania 6, czyli rozkładu trójkątnego. Dodatkowo przyjęto funkcję pomocniczą ograniczającą funkcję gęstości rozkładu, z którego miały zostać wygenerowane liczby losowe przyjęto rozkład równomierny w postaci 10. Oczekiwano, że histogram próbek wygenerowanych przy pomocy metody odrzucania, dla tego przypadku będzie przypominał rozkład trójkątny.

$$f(x) = \begin{cases} \frac{1}{b-a}, & \text{dla } x \in [a, b] \\ 0, & \text{dla } x \notin [a, b] \end{cases} \quad (10)$$

Na rysunku zostały przedstawione wykresy wynikowe dla generowanych próbek



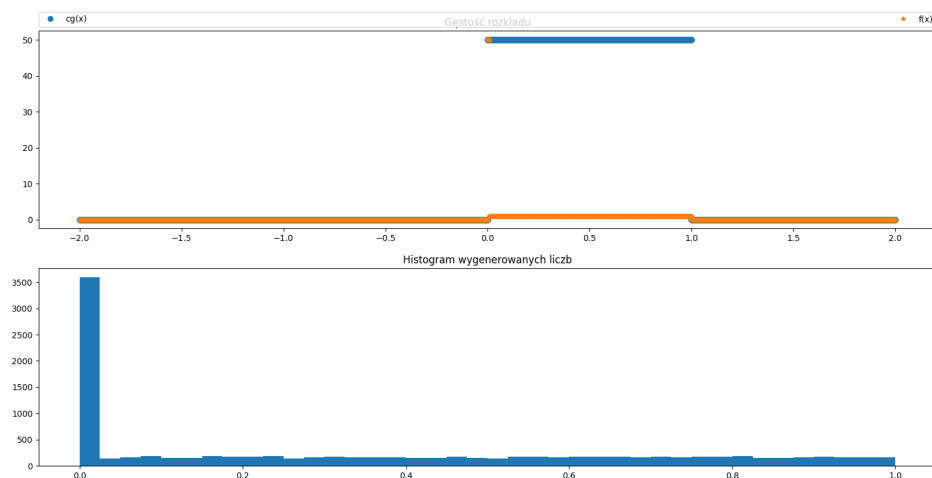
Rysunek 10: Metoda odrzucania zastosowana dla funkcji gęstości rozkładu trójkątnego. Wykresy $f(x)$ oraz $cg(x)$ a także histogram wygenerowanych próbek.

4.2 Rozkład kłopotliwy

Dla kolejnego rozkładu przedstawionego w postaci 11 jako funkcji pomocniczej g użyto również rozkładu normalnego na przedziale $(0, 1]$. W tym przypadku jednak warunkiem wydajnej i właściwej pracy algorytmu jest dobranie odpowiedniego c zapewniającego, że większość próbek nie będzie pobierana z przedziału $(0, \frac{1}{100}]$, aby pokazać wpływ parametru na wynik generowania próbek wybrano współczynnik $c = 1$.

$$f(x) = \begin{cases} 50, & \text{dla } x \in (0, \frac{1}{100}] \\ c, & \text{dla } x \notin (\frac{1}{100}, 1] \end{cases} \quad (11)$$

Na wykresach z rysunku 11 przedstawiono wyniki generowania próbek liczb losowych z wyżej wspomnianego rozkładu.



Rysunek 11: Metoda odrzucania w kłopotliwym przypadku. Wykresy $f(x)$ oraz $cg(x)$ a także histogram wygenerowanych próbek.

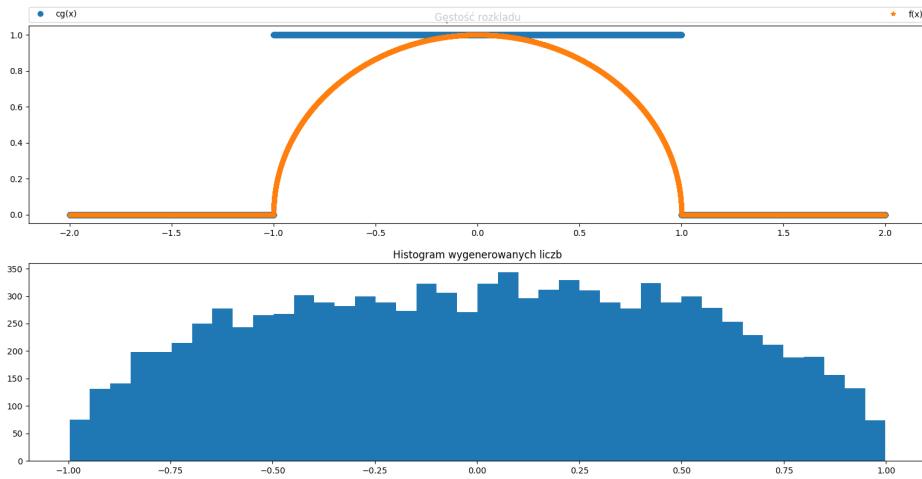
4.3 Rozkład z funkcją gęstości w kształcie półokręgu

Kolejnym przykładem był rozkład, którego wykres funkcji gęstości ma być półokręgiem o danym promieniu. Dodatkowo założeniem jest, aby wartość oczekiwana takiego rozkładu była równa 0. W takim przypadku przyjęto gęstość rozkładu prawdopodobieństwa 12. Dodatkowo w założeniach przyjęto $r = 1$. Z tego powodu wartość oczekiwana wynosi 0, co można zobaczyć w

$$f(x) = \begin{cases} \sqrt{r^2 - x^2}, & \text{dla } x \in [-r, r] \\ 0, & \text{dla } x \notin [-r, r] \end{cases} \quad (12)$$

$$\mathbb{E} X = \int_{-\infty}^{+\infty} xf(x) \rightarrow \mathbb{E} X = \int_{-1}^1 x\sqrt{1-x^2} = \left(-\frac{1}{3}(1-x^2)^{\frac{3}{2}}\right)\Big|_{-1}^1 = 0 \quad (13)$$

Na rysunku 12 widać, że histogram wygenerowanych liczb losowych przy pomocy metody odrzucania przypomina rozkład funkcji f , co potwierdza



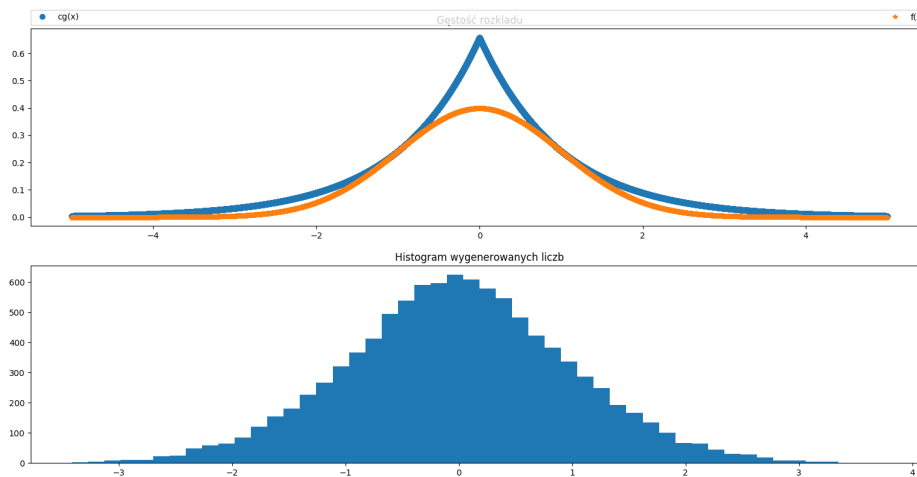
Rysunek 12: Metoda odrzucania zastosowana dla funkcji gęstości w kształcie półokręgu. Wykresy $f(x)$ oraz $cg(x)$ a także histogram wygenerowanych próbek.

4.4 Generowanie liczb z rozkładu normalnego przy pomocy rozkładu Laplace'a

Ostatnim zaimplementowanym generatorem był generator rozkładu normalnego $\mathcal{N}(0, 1)$. Do generowania liczb losowych przyjęto funkcję przedstawioną jako 14. W tym momencie potrzebna jest funkcja, która spełnia zależność 9. Do tego celu wykorzystano generator zaimplementowany na bazie metody odwracania dystrybuantry (8). Można również pokazać, że aby (9) było spełnione to należy przyjąć $c = \sqrt{\frac{2e}{\pi}}$.

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \quad (14)$$

Przy takich założeniach zaimplementowano generator liczb losowych z rozkładu normalnego. Wyniki generowania 10000 próbek w postaci wykresów funkcji oraz histogramu wygenerowanych próbek przedstawiono na rysunku 13.



Rysunek 13: Metoda odrzucania wykorzystana do generowania liczb losowych o rozkładzie normalnym z rozkładu Laplace'a. Wykresy $f(x)$ oraz $cg(x)$ a także histogram wygenerowanych próbek.

5 Wnioski

- Przy tworzeniu generatora z przebiegu pilokształtnego okazało się, że parzysty okres generatora (tzn. ilość żębów") jest szczególnie niekorzystny dla jakości samego generatora.
- Jak widać histogram próbek liczb losowych generowanych z nieznanego rozkładu może dostarczać informacji o samym rozkładzie, gdyż przypomina on funkcję gęstości tego rozkładu.
- Na podstawie badań metody odwrotnej dystrybucyjności można powiedzieć, że jest metoda dość uniwersalna, pozwalająca generować liczby losowe o dowolnym rozkładzie (przy założeniu, że funkcja gęstości rozkładu prawdopodobieństwa jest całkowalna). Dla prostych do odwrócenia funkcji nie widać w niej większych wad, jednak gdyby chcieć uzyskać generator z funkcji, której dystrybucyjność jest niezwykle trudna do odwrócenia zadanie może się znacząco skomplikować.
- Podstawowym problemem powstającym podczas używania metody odrzucania okazuje się poszukiwanie funkcji pomocniczej g ponieważ trzeba je dobierać odpowiednio w zależności od przyjętej funkcji f .
- Dodatkowo przyjęte g dla metody odrzucania powinno „ciasno opasać” gęstość g , dzięki czemu uzyskuje się mniejszą przestrzeń pomiędzy obiema gęstościami, co gwarantuje większe prawdopodobieństwo przyjęcia. Jeżeli funkcja pomocnicza zostanie dobrana nieprawidłowo lub współczynnik c będzie nieodpowiedni proces generowania liczb losowych może się znacząco wydłużyć co również wpływa niekorzystnie na jakość generatora.

Literatura

- [1] Dokumentacja biblioteki matplotlib. <https://matplotlib.org/contents.html>, 03 2019.
- [2] Dokumentacja biblioteki numpy. <https://docs.scipy.org/doc/>, 03 2019.

- [3] Dokumentacja python 3.7.1. <https://docs.python.org/release/3.7.1/>, 03 2019.
- [4] R. S. Jacek Jakubowski. *Wstęp do teorii prawdopodobieństwa*. Jacek Jakubowski, Rafał Sztencel, 2001.
- [5] R. S. Jacek Jakubowski. *Rachunek prawdopodobieństwa dla (prawie) wszystkich*. Jacek Jakubowski, Rafał Sztencel, 2006.
- [6] Dr hab. inż Grzegorz Mzyk prof PWr. Generacja liczb losowych o różnych rozkładach. <http://staff.iiar.pwr.wroc.pl/grzegorz.mzyk/kmi/kmi03.pdf>, 03 2019.
- [7] R. Z. Robert Wieczorkowski. *Komputerowe generatory liczb losowych*. Wydawnictwa Naukowo-Techniczne, 1997.
- [8] K. Sigman. Acceptance-rejection method. <http://www.columbia.edu/~ks20/4703-Sigman/4703-07-Notes-ARM.pdf>, 03 2019.