

Symulacje Komputerowe

Raport: 1

Temat sprawozdania **Generowanie zmiennych losowych**
Nazwisko i Imię prowadzącego kurs **dr Michał Balcerk**

Wykonawca:	
Imię i Nazwisko, nr indeksu	Kacper Budnik, 262286 Szymon Malec, 262276
Wydział	Wydział matematyki, W13
Termin zajęć:	Wtorek, 11 ¹⁵
Numer grupy ćwiczeniowej	T00-70d
Data oddanie sprawozdania:	10 maja 2022
Ocena końcowa	

Adnotacje dotyczące wymaganych poprawek oraz daty otrzymania poprawionego sprawozdania

1. Wstęp

Cel sprawozdania był dobrze nam znany
Sposób generowania zmiennych miał być pokazany
By w przyszłości symulacje wielkie tworzyć
I pieniążki tym samym szybko mnożyć.

Zanim pokażemy co tygryski lubią najbardziej
Omówimy przebieg naszej pracy starannej.
By przyjemnie i łatwo się Panu raport czytało
Postaram się przedstawić co się w nim znalazło.

Rozpoczęliśmy od generatora kongruentnego stworzenia
Oraz sprawdzenia na wykresie jego zagęszczenia
By pokazać niezależność prób generowanych
W zależności od parametrów starannie dobranych

Oczywiście opisaliśmy wszystkie generowania metody
Bo na wykładzie były przedstawione każdej dowody.
I nawet porównaliśmy dwa sposoby ze sobą
By wiedzieć który możemy obdarzyć koroną.

W opisie przedstawiliśmy informacje ogólne,
Które z wykładami zdecydowanie mają części wspólne.
Szczególnie algorytm do generowania zmiennych potrzebny
Ponieważ przedstawiony został nam w sposób chwalebny.

Naturalnie przetestowaliśmy poprawność metod pokazanych.
Sprawdziliśmy czy histogram do gęstości pasuje dobranych,
Czy dystrybuanta empiryczna z teoretyczną się pokrywa,
Czy na qq-plotie linia płaszczyznę na dwie połowy rozrywa.

By powiedzieć, że nasze dzieło jest pełni kompletne
Jeszcze nam zakończenie było do szczęścia potrzebne.
Tym samym całe dzieło opisane dokładnie zostało
I nadzieję mamy, że zachwytu dostarczy Panu nie mało.

2. Liniowy generator kongruentny^[1]

Jest to narzędzie służące do generowania liczb pseudolosowych. Zanim omówimy jego działanie, wyjaśnijmy czym są wspomniane liczby pseudolosowe. Są to liczby przypominające losowe, jednakże są one wygenerowane według pewnego schematu. W tym przypadku są one obliczane według wzoru

$$x_n = (ax_{n-1} + c) \bmod m,$$

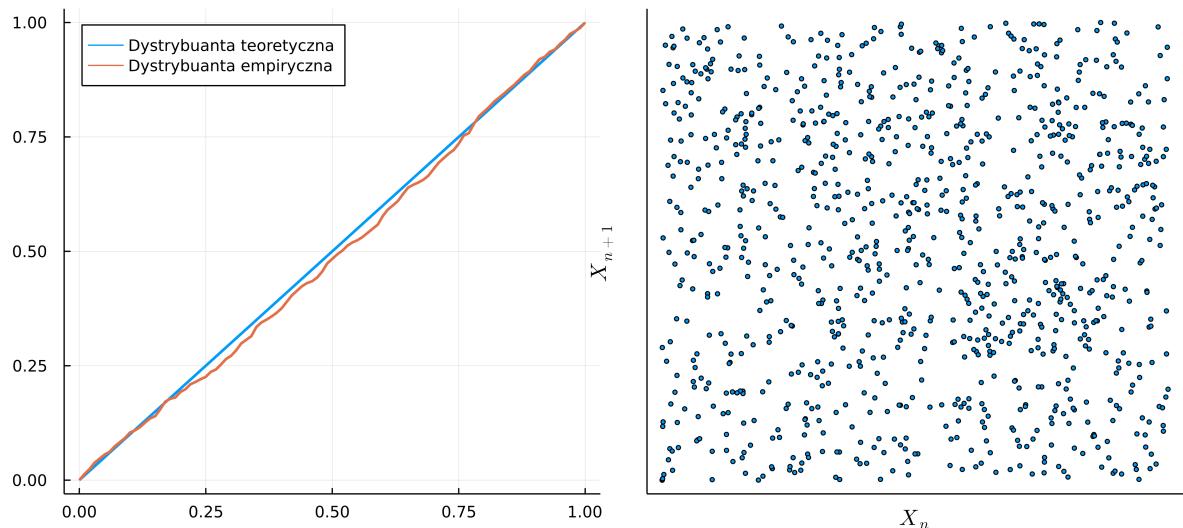
gdzie $a, c, m \in \mathbb{Z}_+$ muszą spełniać $0 < a < m$ i $0 \leq c < m$ oraz $0 \leq x_0 < m$. Liczbę x_0 nazywamy ziarnem, na podstawie którego wyliczane są kolejne wartości x_n . Jak się okazuje, wartości $\frac{x_n}{m}$ posiadają własności zbliżone do rozkładu jednostajnego $\mathcal{U}(0, 1)$.

Gdy już znamy wzór generatora, pojawia się pytanie jakie wartości współczynników a, b, c dobrać, by wygenerowane liczby jak najlepiej przypominały liczby losowe. Istnieje wiele popularnych wyborów. Przykłady:

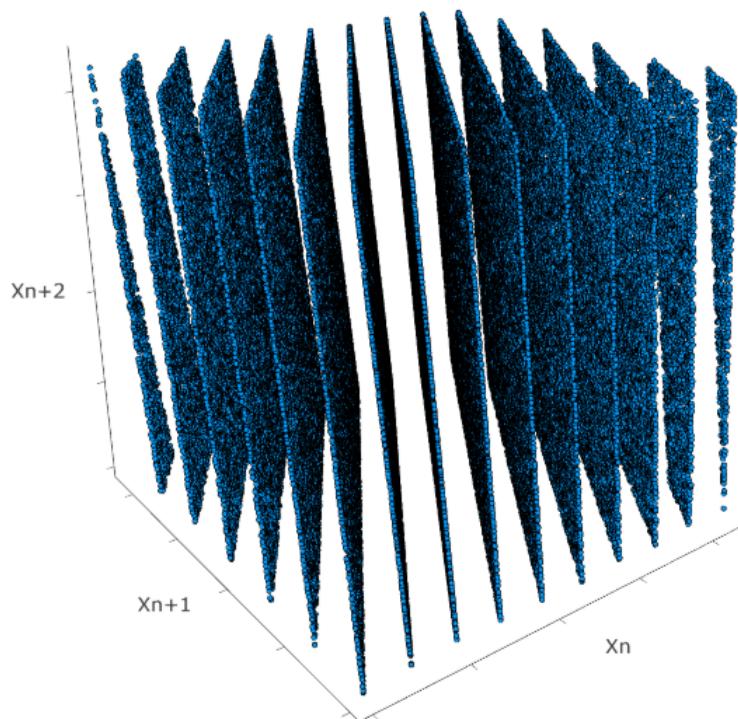
- **generator RANDU^[2]:** $a = 2^{16} + 3$, $c = 0$, $m = 2^{31}$

Jest to jeden z najgorszych w historii generatorów kongruentnych. Był wykorzystywany głównie w latach 60-tych i 70-tych. Wybierzmy $x_0 = 1$ i wykonajmy kilka testów.

Rysunek 1: Po lewej porównanie dystrybuanty empirycznej z teoretyczną rozkładu jednostajnego $\mathcal{U}(0, 1)$, po prawej test 2D. Oba testy przeprowadzone zostały dla 1000 liczb wygenerowanych generatorem RANDU.



Rysunek 2: Test 3D dla 100 000 liczb.

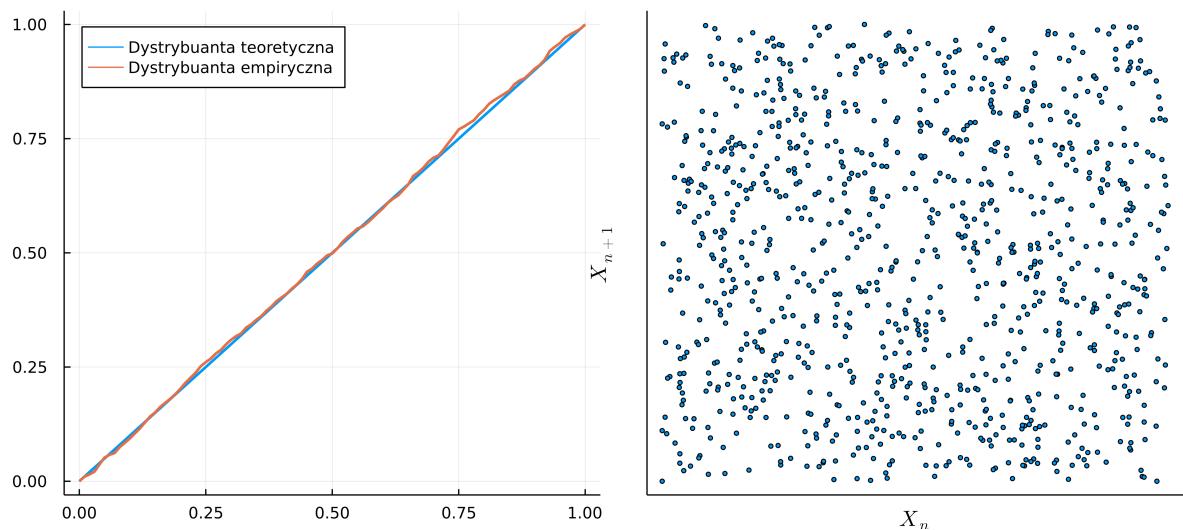


Widzimy, że dystrybuanta empiryczna pokrywa się z teoretyczną. Z kolei na teście 2D punkty wydają się być rozłożone losowo. Jednakże test 3D udowadnia, że liczby te są wygenerowane na podstawie schematu, ponieważ widać wyraźnie, że punkty układają się w uporządkowany sposób. Wniosek jest taki, że generator ten nie jest najlepszy.

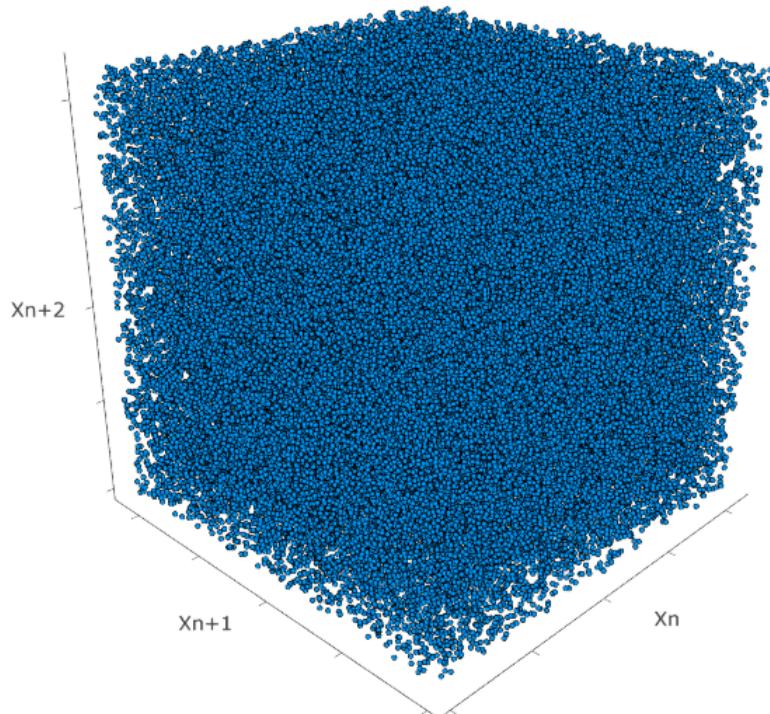
- **generator MTH\$RANDOM**^[2]: $a = 69069$, $c = 5$, $m = 2^{32}$

Ponownie wybierzmy $x_0 = 1$ i wykonajmy te same testy.

Rysunek 3: Po lewej porównanie dystrybuanty empirycznej z teoretyczną rozkładu jednostajnego $\mathcal{U}(0, 1)$, po prawej test 2D. Oba testy przeprowadzone zostały dla 1000 liczb wygenerowanych generatorem MTH\$RANDOM.



Rysunek 4: Test 3D dla 100 000 liczb.



Jak możemy zauważyć, podobnie jak w poprzednim przykładzie, dystrybuanty empiryczna i teoretyczna są do siebie zbliżone oraz rozłożenie punktów w teście 2D wygląda na losowe. Dodatkowo test 3D także sugeruje niezależne rozłożenie punktów. Na tej podstawie możemy stwierdzić, że generator MTH\$RANDOM jest zdecydowanie lepszy od generatora RANDU.

3. Metoda odwrotnej dystrybuanty

3.1. Opis

Metoda ta polega na generowaniu zmiennej losowej X generując zmienną U z rozkładu jednoznacznego oraz nałożeniu na nią funkcji odwrotnej dystrybuanty.

Algorytm dla rozkładów dyskretnych^[3]

Załóżmy, że rozkład X ma postać $P(X = x_i) = p_i$, $i = 1, 2, \dots$.

1. Generuj $U \sim \mathcal{U}(0, 1)$.
2. Wyznacz $j \in \mathbb{N}$ takie, że $\sum_{i=1}^{j-1} p_i < U \leq \sum_{i=1}^j p_i$.
3. Zwróć $X = x_i$.

Algorytm dla rozkładów ciągłych^[4]

Załóżmy, że X ma dystrybuantę $F(x)$.

- a) Jeśli dystrybuanta jest ścisłe rosnąca:
 1. Generuj $U \sim \mathcal{U}(0, 1)$.
 2. Zwróć $X = F_X^{-1}(U)$.
- b) Jeśli dystrybuanta nie jest ścisłe rosnąca:
 1. Generuj $U \sim \mathcal{U}(0, 1)$.
 2. Zwróć $X = \tilde{F}_X^{-1}(U)$, gdzie $\tilde{F}_X^{-1}(y) = \inf\{x \in \mathbb{R} : F_X(x) \geq y\}$.

3.2. Przykłady

Dyskretny - rozkład Poissona

Celem jest wygenerowanie $X \sim \text{Poiss}(\lambda)$, $\lambda > 0$. Przypomnijmy jak wygląda rozkład Poissona:

$$p_i = P(X = i) = e^{-\lambda} \frac{\lambda^i}{i!}.$$

Zdefiniujmy

$$s_j = \sum_{i=0}^j p_i.$$

Algorytm będzie więc polegał na generowaniu $U \sim \mathcal{U}(0, 1)$ i wyznaczaniu $j \in \mathbb{N}$ takiego, że

$$s_{j-1} < U \leq s_j.$$

To znaczy, że musimy sprawdzać kolejne przedziały $(s_{j-1}, s_j]$, czy wpada do nich U . W przypadku, gdy λ , czyli średnia wartość rozkładu zmiennej X , jest duża, przechodzenie przez wszystkie przedziały może zajmować sporo czasu. Rozpatrzmy właśnie taki przypadek. Żeby skrócić czas wykonania, możemy zacząć sprawdzanie od przedziału $(s_{\lceil \lambda \rceil - 1}, s_{\lceil \lambda \rceil}]$. Następnie,

jeśli U jest mniejsze, to sprawdzamy przedziały na lewo, jeśli większe to na prawo. Dodatkowo zauważmy, że

$$p_{i+1} = p_i \cdot \frac{\lambda}{i+1}.$$

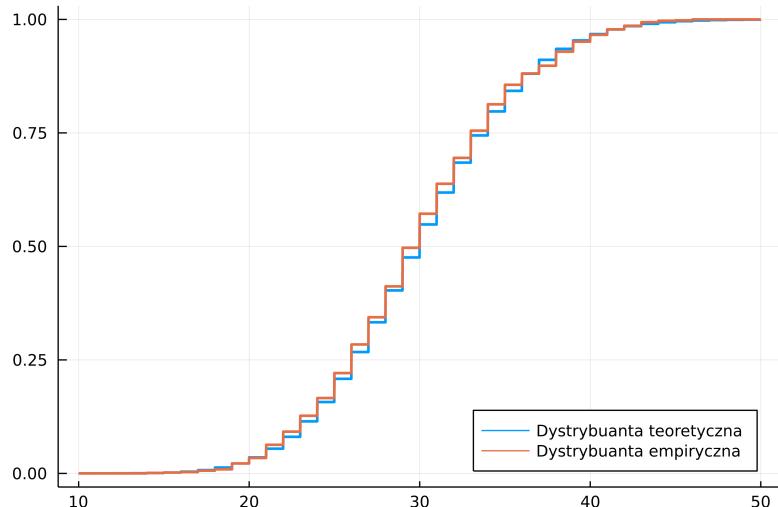
Stąd kolejne p_i otrzymamy domnażając $\frac{\lambda}{i+1}$. Na podstawie tych informacji możemy zbudować algorytm:

1. Oblicz wartości p_i oraz s_i dla $i = 0, 1, \dots, \lceil \lambda \rceil$.
2. Zdefiniuj $j = \lceil \lambda \rceil$, $p = p_{\lceil \lambda \rceil}$, $s = s_{\lceil \lambda \rceil}$.
3. Generuj $U \sim \mathcal{U}(0, 1)$.
4. Jeśli $U > s$:
 - a) $p = p \cdot \frac{\lambda}{j+1}$
 $s = s + p$
 $j = j + 1$
 - b) Jeżeli $U < s$, zwróć $X = j$. W przeciwnym razie wróć do a).

Jeśli $U < s$:

- a) $s = s_{j-1}$
 $j = j - 1$
- b) Jeżeli $U > s$, zwróć $X = j + 1$. W przeciwnym razie wróć do a).

Rysunek 5: Porównanie dystrybuanty empirycznej i teoretycznej dla próby z rozkładu $\text{Poiss}(30)$ o długości $n = 1000$.



Ciągły - rozkład Cauchy'ego

Chcemy wygenerować $X \sim \mathcal{C}(\mu, \sigma)$. Dystrybuanta rozkładu Cauchy'ego ma postać

$$F(x) = \frac{1}{\pi} \arctan\left(\frac{x - \mu}{\sigma}\right) + \frac{1}{2}.$$

Za pomocą elementarnych przekształceń jesteśmy w stanie otrzymać funkcję odwrotną

$$F^{-1}(y) = \sigma \tan\left(\pi\left(y - \frac{1}{2}\right)\right) + \mu.$$

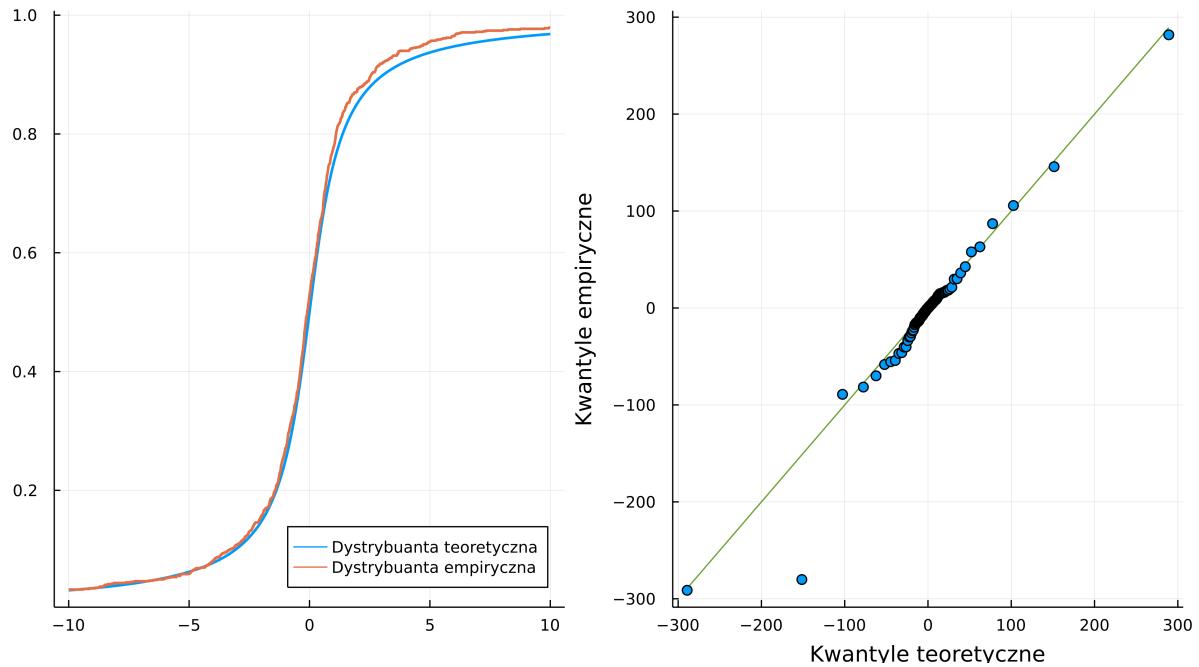
Zatem, żeby wygenerować X , należy najpierw wygenerować $U \sim \mathcal{U}(0, 1)$ i zwrócić $F^{-1}(U)$. Możemy to jednak łatwo uprościć. Niech $V \sim \mathcal{U}\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. Wtedy

$$V \stackrel{d}{=} \pi\left(U - \frac{1}{2}\right).$$

Ostatecznie algorytm będzie wyglądał następująco:

1. Generuj $V \sim \mathcal{U}\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$.
2. Zwróć $X = \sigma \tan(V) + \mu$.

Rysunek 6: Porównanie dystrybuanty empirycznej i teoretycznej oraz wykres kwantylowy dla próby z rozkładu $\mathcal{C}(0, 1)$ o długości $n = 1000$.



4. Metoda akceptacji i odrzucenia

4.1. Opis metody dla przypadku dyskretnego^[5]

Metoda akceptacji i odrzucenia służy do generowania zmiennej losowej X przy użyciu innych zmiennych. By móc wykorzystać tą metodę, muszą być spełnione:

- potrafimy efektywnie generować inną zmienną losową Y ,
- zmienne X oraz Y muszą być skupione na tym samym zbiorze,
- potrafimy wyznaczyć stałą c taką że $\frac{P(X = i)}{P(Y = i)} = \frac{p_i}{q_i} \leq c$ dla każdego i .

Jeśli są spełnione powyższe założenia, możemy użyć poniższego algorytmu do generowania zmiennej X .

Algorytm

1. Generuj jedną realizację Y .
2. Generuj $U \sim \mathcal{U}(0, 1)$, $U \perp\!\!\!\perp Y$.
3. Jeśli $U \leq \frac{p_Y}{cq_Y}$ zwróć $X = Y$, w przeciwnym wróć do 1.

4.2. Opis metody dla przypadku ciągłego^[6]

Metoda akceptacji i odrzucenia służy do generowania zmiennej losowej X o gęstości $f(x)$ przy użyciu innych zmiennych. By móc wykorzystać tą metodę, muszą być spełnione:

- potrafimy efektywnie generować inną zmienną losową Y o gęstości $g(x)$,
- zmienne X oraz Y muszą być skupione na tym samym zbiorze,
- potrafimy wyznaczyć stałą c taką że $\sup \frac{f(x)}{g(x)} \leq c$ dla każdego x , gdzie $g(x) \neq 0$.

Algorytm

1. Generuj jedną realizację Y
2. Generuj $U \sim \mathcal{U}(0, 1)$, $U \perp\!\!\!\perp Y$
3. Jeśli $U \leq \frac{f(Y)}{cg(Y)}$ zwróć $X = Y$, w przeciwnym wróć do 1.

4.3. Szybkość algorytmu

W obu przypadkach, ciągłym i dyskretnym, prawdopodobieństwo, że zmienna zostanie zaakceptowana wynosi $\frac{1}{c}$. Liczba iteracji algorytmu potrzebnych do wygenerowania zmiennej ma rozkład $\text{Geom}(\frac{1}{c})$, zatem średnia liczba iteracji potrzebna do wygenerowania tej zmiennej to c . Z tego powodu powinniśmy dobrać ją możliwie jak najmniejszą. Najoptimalniej wybrać $c = \max \frac{p_Y}{q_Y}$ w przypadku dyskretnym oraz $c = \sup \frac{f(x)}{g(x)}$ dla przypadku ciągłego.

4.4. Przykłady

Dyskretny

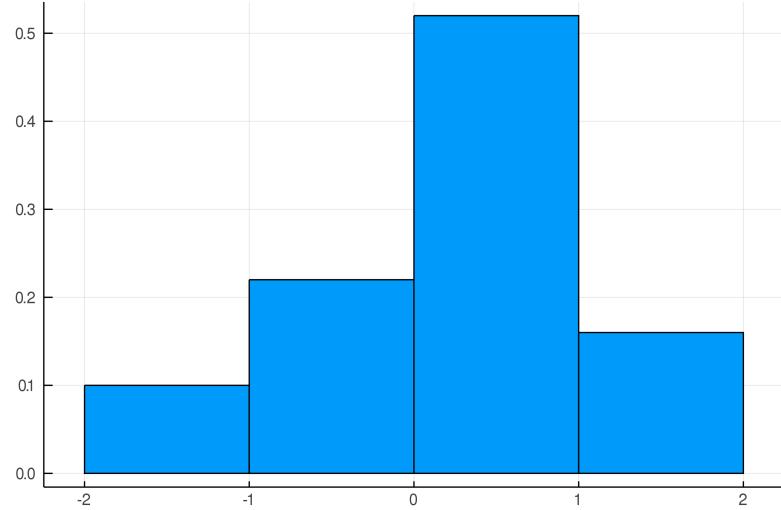
Niech X ma następujący rozkład

$$P(X = i) = \begin{cases} 0.1, & \text{dla } i = -2 \\ 0.2, & \text{dla } i = -1 \\ 0.5, & \text{dla } i = 0 \\ 0.2, & \text{dla } i = 1 \end{cases}$$

Do generowania tej zmiennej potrzebujemy zmienną, która jest skupiona na tym samym zbiorze. Najprostszym pomysłem jest wykorzystanie zmiennej z rozkładu jednostajnego dyskretnego $Y \sim \mathcal{D}\mathcal{U}(-2, -1, 0, 1)$. Prawdopodobieństwo, że Y przyjmie daną wartość stale wynosi $\frac{1}{4}$, zatem łatwo wyliczyć stałą $c = 4 \max_i P(X = i) = 4 \cdot 0.5 = 2$. Znając tą stałą wystarczy zastosować dany algorytm:

1. Generuj jedną realizację $Y \sim [4 \cdot \mathcal{U}(0, 1)] - 2$
2. Generuj $U \sim \mathcal{U}(0, 1)$, $U \perp\!\!\!\perp Y$
3. Jeśli $U \leq 2p_Y$ zwróć $X = Y$, w przeciwnym wróć do 1.

Rysunek 7: Sprawdzenie poprawności metody Akceptacji-Odrzucenie dla próby o rozmiarze $n = 100$



Ciągły

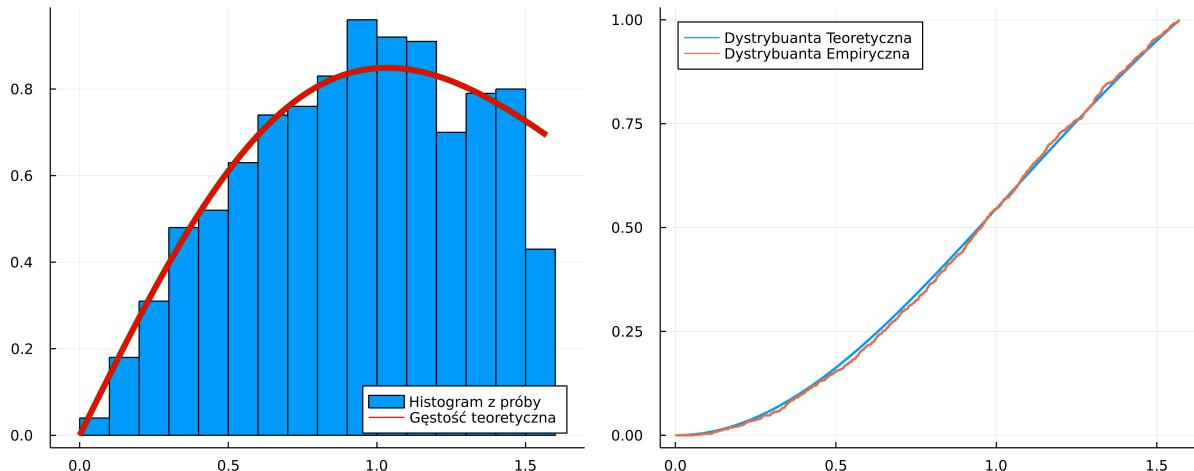
Chcemy wygenerować zmienną X o gęstości $f(x) = \ln(2) \cdot \sin(x) \cdot 2^{\cos(x)} 1_{(0, \frac{\pi}{2})}$. Użyjemy do tego zmiennej $Y \sim \mathcal{U}(0, \frac{\pi}{2})$ o funkcji gęstości $g(x) = \frac{2}{\pi} 1_{(0, \frac{\pi}{2})}$. Zaczniemy od wyliczenia stałej c .

$$c = \sup_{x \in (0, \frac{\pi}{2})} \frac{f(x)}{g(x)} = \sup_{x \in (0, \frac{\pi}{2})} \sin(x) \cdot 2^{\cos(x)} \cdot \frac{\pi \ln(2)}{2} \approx \frac{4}{3}$$

Teraz wystarczy się stosować do następującego algorytmu, czyli

1. Generuj $Y \sim \frac{\pi}{2} \mathcal{U}(0, 1)$
2. Generuj $U \sim \mathcal{U}(0, 1)$, $U \perp\!\!\!\perp Y$
3. Jeśli $U \leq \sin(Y) \cdot 2^{\cos(Y)} \cdot \frac{3\pi \ln(2)}{8}$ zwróć $X = Y$, w przeciwnym wróć do 1.

Rysunek 8: Sprawdzenie poprawności metody Akceptacji-Odrzucenie dla próby o rozmiarze $n = 1000$



5. Metoda splotowa^[7]

5.1. Opis

Metoda ta pozwala wygenerować pewną zmienną losową X , przy pomocy innych zmiennych, które potrafimy efektywnie generować, i zsumowaniu ich. Założymy, że

$$X \stackrel{d}{=} Y_1 + Y_2 + \cdots + Y_n,$$

gdzie Y_i to zmienne losowe niezależne. Wtedy algorytm wygląda następująco:

1. Generuj Y_1, Y_2, \dots, Y_n .
2. Zwróć $X = Y_1 + Y_2 + \cdots + Y_n$.

5.2. Przykłady

Dyskretny - rozkład dwumianowy

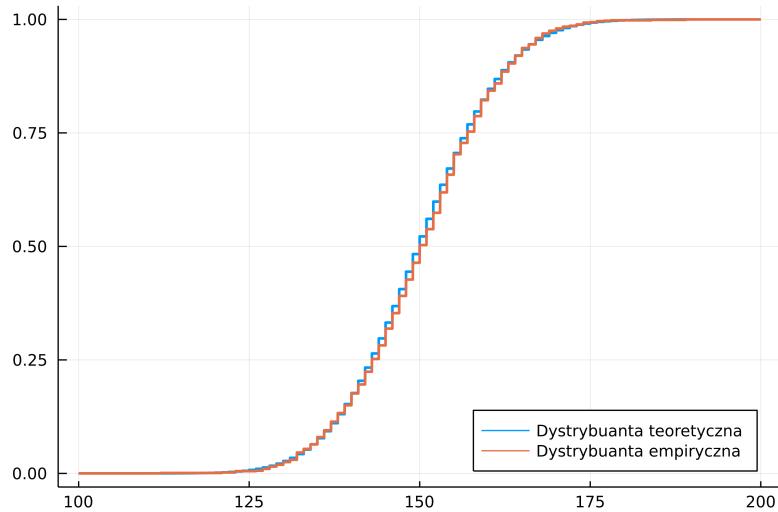
Niech $X \sim \mathcal{B}(n, p)$ oraz niech Y_1, Y_2, \dots, Y_n będzie ciągiem niezależnych zmiennych losowych będących wynikami prób Bernoulliego z prawdopodobieństwem sukcesu p . Wtedy

$$X \stackrel{d}{=} Y_1 + Y_2 + \cdots + Y_n.$$

Stąd mamy algorytm:

1. Generuj Y_1, Y_2, \dots, Y_n , gdzie $Y_i \sim \mathcal{B}(1, p)$.
2. Zwróć $X = Y_1 + Y_2 + \cdots + Y_n$.

Rysunek 9: Porównanie dystrybuanty empirycznej i teoretycznej dla próby z rozkładu $\mathcal{B}(500, \frac{3}{10})$ o długości $n = 1000$.



Ciągły - rozkład chi-kwadrat

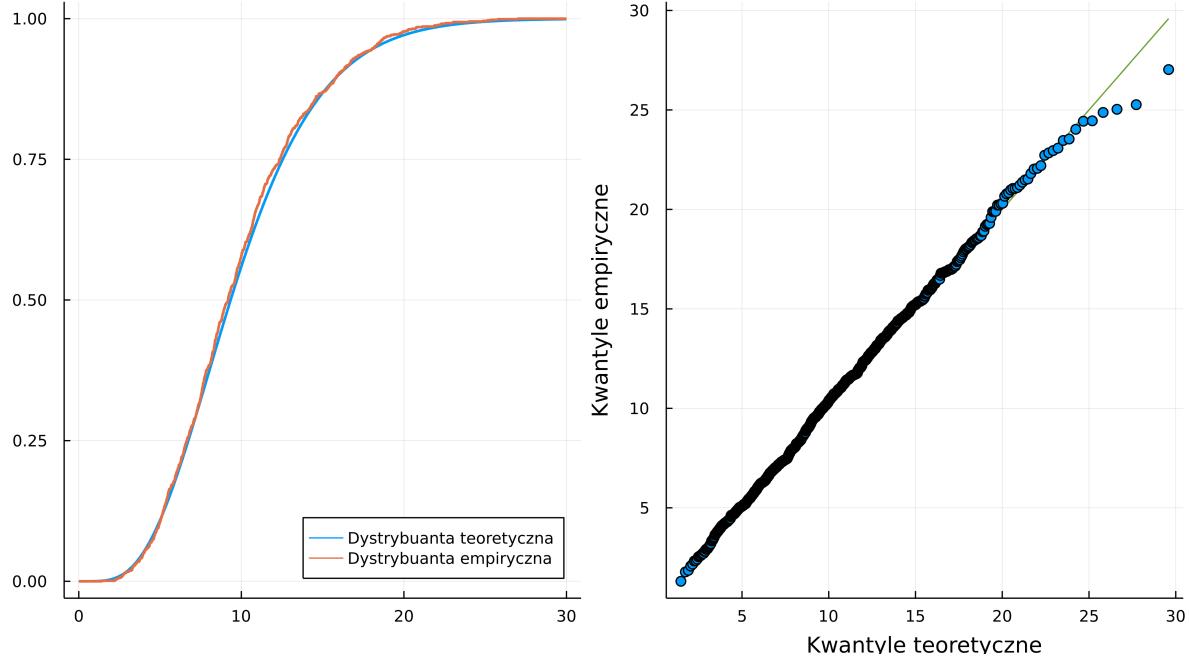
Niech $X \sim \chi^2(n)$ oraz niech Y_1, Y_2, \dots, Y_n będzie ciągiem niezależnych zmiennych losowych o rozkładzie normalnym standardowym. Wtedy

$$X \stackrel{d}{=} Y_1^2 + Y_2^2 + \cdots + Y_n^2.$$

Stąd otrzymujemy algorytm:

1. Generuj Y_1, Y_2, \dots, Y_n , gdzie $Y_i \sim \mathcal{N}(0, 1)$.
2. Zwróć $X = Y_1^2 + Y_2^2 + \dots + Y_n^2$.

Rysunek 10: Porównanie dystrybuanty empirycznej i teoretycznej oraz wykres kwantylowy dla próby z rozkładu $\chi^2(10)$ o długości $n = 1000$.



6. Metoda kompozycji^[8]

6.1. Opis

Metoda ta służy do generowania wyłącznie ciągłych rozkładów. Założymy, że zmienna losowa X , którą chcemy symulować, ma dystrybuantę postaci

$$F(x) = \sum_{i=1}^n p_i F_i(x),$$

gdzie $p_i > 0$, $\sum_{i=1}^n p_i = 1$ oraz F_i to dystrybuanty pewnych zmiennych losowych Y_i . Jeżeli zmienne Y_i mają gęstość, to równoważne będzie założenie, że gęstość zmiennej X ma postać

$$f(x) = \sum_{i=1}^n p_i f_i(x),$$

gdzie f_i są gęstościami zmiennych Y_i . Do wygenerowania X może posłużyć poniższy algorytm:

1. Generuj zmienną losową I o rozkładzie $P(I = i) = p_i$.
2. Generuj Y_I .
3. Zwróć $X = Y_I$.

6.2. Przykład

Chcemy wygenerować zmienną losową X o gęstości

$$f(x) = \frac{1}{3\pi(1+x^2)} + \frac{2}{3\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

Jeśli zapiszemy ją w postaci

$$f(x) = \frac{1}{3} \left(\frac{1}{\pi(1+x^2)} \right) + \frac{2}{3} \left(\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \right),$$

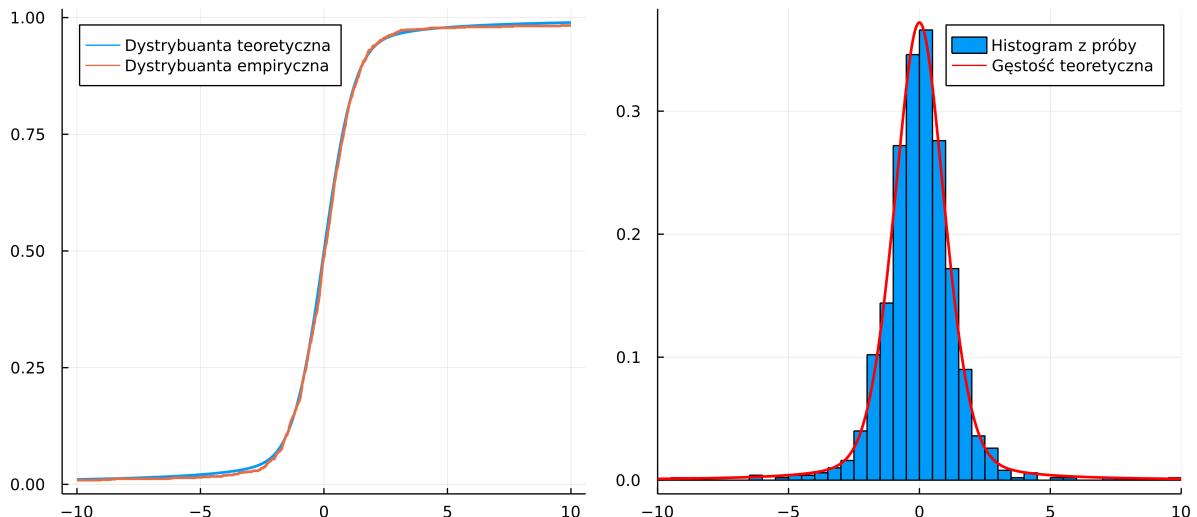
to możemy zauważyć, że jest to suma gęstości rozkładów Cauchy'ego i normalnego standardego. Weźmy zatem $Y_1 \sim \mathcal{C}(0, 1)$ oraz $Y_2 \sim \mathcal{N}(0, 1)$. Niech $p_1 = \frac{1}{3}$ i $p_2 = \frac{2}{3}$. Wtedy możemy zapisać

$$f(x) = p_1 f_1(x) + p_2 f_2(x),$$

gdzie f_1 i f_2 to gęstości odpowiednio Y_1 i Y_2 . Stąd algorytm prezentuje się następująco:

1. Generuj I o rozkładzie $P(I = 1) = \frac{1}{3}$, $P(I = 2) = \frac{2}{3}$.
2. Generuj Y_I .
3. Zwróć $X = Y_I$.

Rysunek 11: Porównanie histogramu z gęstością teoretyczną oraz porównanie dystrybuanty empirycznej i teoretycznej dla próby z powyższego rozkładu mieszanego o długości $n = 1000$.



7. Metoda Boxa-Mullera^[9]

7.1. Opis

Najważniejszym rozkładem w prawdopodobieństwie jest rozkład normalny. Dlatego istnieje specjalna metoda jego generowania. Łatwo można pokazać (co zrobiliśmy na kursie Modelowanie Stochastyczne), że jeśli U_1, U_2 - iid, $U_1 \sim \mathcal{U}(0, 1)$, to zmienne

$$\begin{aligned} X &= \sqrt{-2 \ln(U_1)} \cos(2\pi U_2) \\ Y &= \sqrt{-2 \ln(U_1)} \sin(2\pi U_2) \end{aligned}$$

są niezależne oraz obie mają rozkład normalny standardowy $\mathcal{N}(0, 1)$. Na tej podstawie powstał poniższy algorytm

Algorytm

1. Generuj zmienne U_1, U_2 - iid, $U_1 \sim \mathcal{U}(0, 1)$
2. Zwróć:

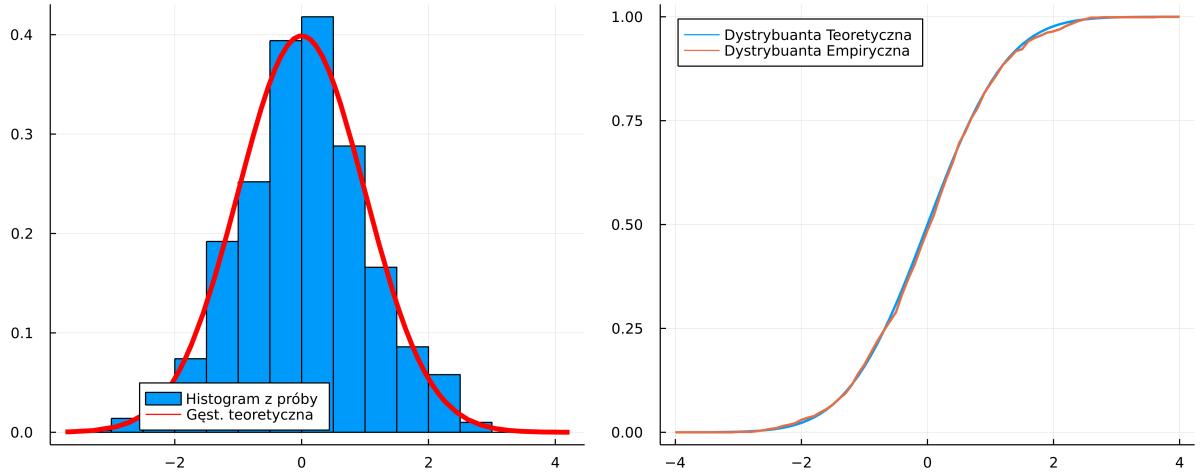
$$X = \sqrt{-2 \ln(U_1)} \cos(2\pi U_2)$$

$$Y = \sqrt{-2 \ln(U_1)} \sin(2\pi U_2)$$

Algorytm ten generuje od razu dwie próby z rozkładu normalnego standardowego. By wygenerować zmienną z rozkładu $\mathcal{N}(\mu, \sigma^2)$, wystarczy wcześniej wygenerowane zmienne pomnożyć przez σ po czym dodać do nich μ .

7.2. Przykłady

Rysunek 12: Sprawdzenie poprawności metody Boxa-Mullera dla próby o rozmiarze $n = 1000$



8. Metoda biegunkowa^[10]

8.1. Opis

Jak wspomnieliśmy poprzednio, rozkład normalny jest najważniejszym rozkładem prawdopodobieństwa. Z tego powodu jest on bardzo często wykorzystywany, ważne jest by generowanie próby trwało jak najkrócej. Niestety w metodzie Boxa-Mullera pojawiają się funkcje sin oraz cos. Wyznaczenie ich wartości jest stosunkowo czasochłonne dla procesora, dlatego poszukiwana była metoda, która nie zawiera tych funkcji. Przekształcając wzór wykorzystywany w metodzie Boxa-Mullera, można pokazać, że jeśli wektor (V_1, V_2) ma rozkład jednostajny na kole jednostkowym, to zmienne

$$X = \sqrt{\frac{-2 \ln R^2}{R^2}} V_1$$

$$Y = \sqrt{\frac{-2 \ln R^2}{R^2}} V_2$$

mają rozkład normalny standardowy $\mathcal{N}(0, 1)$ oraz $X \perp\!\!\!\perp Y$. Ponieważ umiemy w sposób efektywny generować wektor losowy (V_1, V_2) , wystarczy skorzystać z poniższego algorytmu

Algorytm

1. Generuj zmienne V_1, V_2 - iid, $V_1 \sim \mathcal{U}(-1, 1)$
2. Wyznacz $R^2 = V_1^2 + V_2^2$
3. Jeżeli $R^2 \geq 1$ wróć do 1.
4. Zwróć

$$X = \sqrt{\frac{-2 \ln R^2}{R^2}} V_1$$

$$Y = \sqrt{\frac{-2 \ln R^2}{R^2}} V_2$$

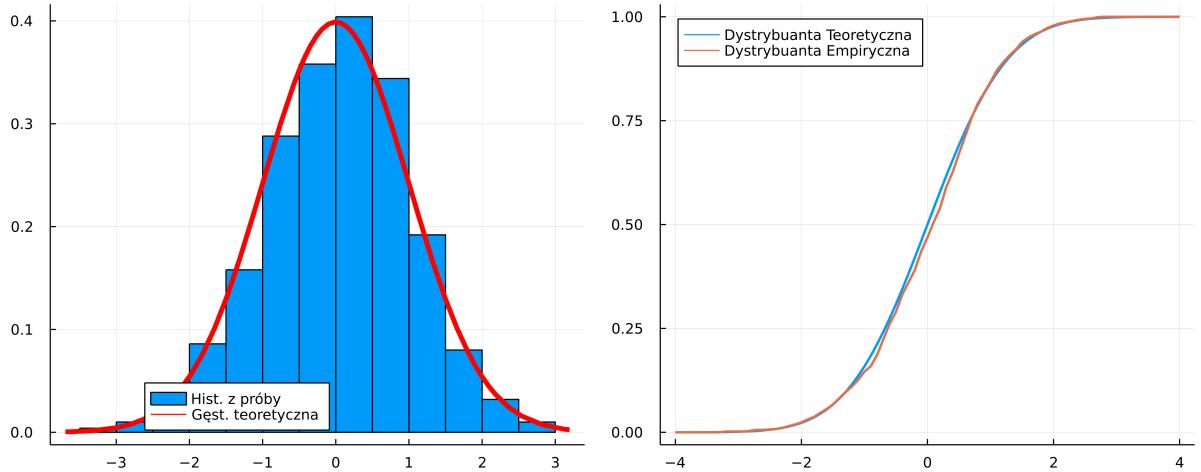
Podobnie jak w metodzie Boxa-Mullera, algorytm ten zwraca dwie zmienne niezależne z rozkładu normalnego, więc by otrzymać zmienne z rozkładu $\mathcal{N}(\mu, \sigma^2)$, tak samo jak wcześniej, mnożymy wygenerowaną próbę przez σ i dodajemy do niej μ .

8.2. Szybkość algorytmu

Do wygenerowania dwóch prób rozkładu normalnego korzystając z tego algorytmu potrzebne jest powtórzenie go średnio $\frac{4}{\pi} \approx 1.25$ razy. Pomimo, że algorytm będzie czasem wykonywany wielokrotnie, to poprzez pozbycie się funkcji trygonometrycznych czas rzeczywisty wygenerowania danej próby będzie krótszy niż w poprzedniej metodzie, co pokażemy później.

8.3. Przykłady

Rysunek 13: Sprawdzenie poprawności metody biegunkowej dla próby o rozmiarze $n = 1000$



9. Porównanie metod generowania rozkładu normalnego

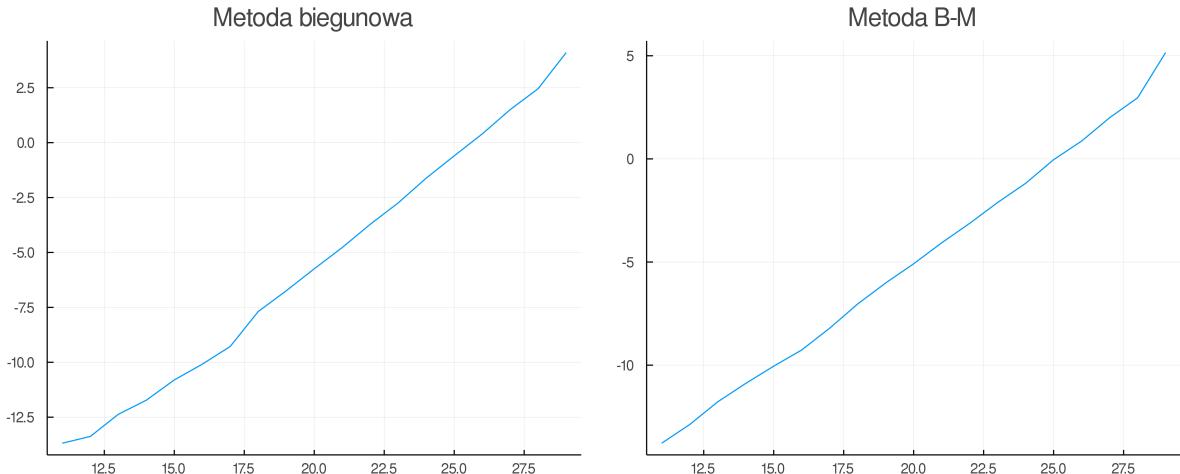
Dwie ostatnie metody, jakie pokazaliśmy, dotyczą generowania tego samego rozkładu, dlatego ważne wydaje się, by porównać czas ich działania. Wykonaliśmy pomiary czasu generowania różnych wielkości prób (o wielkości N) dla metody biegunkowej oraz metody Boxa-Mullera (B-M). Dodatkowo policzyliśmy stosunek czasu potrzebnego do wygenerowania metodą pierwszą do metody B-M. W poniższej tabeli zamieściliśmy średni czas generowania próby. Dla wartości do 2^{24} pomiar wykonaliśmy 1000 razy, dla liczb nie większych niż 2^{27} zmniejszyliśmy pomiar do 200 prób, a pozostałe wielkości prób wygenerowaliśmy jedynie 10 razy.

Tabela 1: Czas wykonania danych metod dla danych wielkości prób

N	Biegunowa	B-M	Stosunek
2^{11}	0.000076	0.000071	1.070
2^{12}	0.000094	0.000133	0.707
2^{13}	0.000189	0.000285	0.663
2^{14}	0.000295	0.000529	0.558
2^{15}	0.000559	0.000944	0.592
2^{16}	0.000919	0.001627	0.565
2^{17}	0.001612	0.003385	0.476
2^{18}	0.004848	0.007636	0.635
2^{19}	0.009334	0.015469	0.603
2^{20}	0.018661	0.029472	0.633
N	Biegunowa	B-M	Stosunek
2^{20}	0.019	0.030	0.633
2^{21}	0.037	0.060	0.614
2^{22}	0.076	0.115	0.663
2^{23}	0.149	0.231	0.647
2^{24}	0.326	0.440	0.742
2^{25}	0.659	0.974	0.677
2^{26}	1.316	1.821	0.723
2^{27}	2.836	3.990	0.711
2^{28}	5.500	7.786	0.706
2^{29}	17.157	35.594	0.482

Możemy przedstawić jeszcze te wartości na wykresach. Ponieważ wielkości naszej próby rosną wykładniczo na obie osie nałożyliśmy \log_2 .

Rysunek 14: Wykres czasu generowania zmiennych w zależności od wielkości próby



Widzimy, że oba wykresy przypominają linię prostą, zatem możemy przypuszczać, że nasze algorytmy mają złożoność czasową rzędu $o(x^n)$, gdzie n jest współczynnikiem kierunkowym powyższych wykresów. Z niewielkimi odchyleniami możemy obliczyć, że $n \approx 1$, z czego wynika, że czas działania obu algorytmów rośnie liniowo wraz z wielkością próby.

Możemy odczytać z tabeli, że metoda biegunowa jest znaczco szybsza. Na podstawie tych danych widzimy, że tym sposobem możemy wygenerować tę samą próbę poświęcając średnio 0.633 czasu, który musielibyśmy poświęcić korzystając z metody Boxa-Mullera.

10. Podsumowanie

Zgodnie z celem raportu przedstawiliśmy wszystkie znane nam metody generowania zmiennych losowych. Każdą z metod opisaliśmy poprzez przedstawienie ogólnej idei stojącej za nią, założeń (jeśli jakieś były) kiedy możemy z danej metody skorzystać oraz algorytmu jej działania. Dodatkowo sprawdziliśmy poprawność metod poprzez wygenerowanie niewielkiej próby z danego rozkładu i porównaliśmy jej zgodność z rozkładem teoretycznym na wykresach. O poprawności działania algorytmów świadczy to, że:

1. dystrybuanta empiryczna pokrywa się z teoretyczną,
2. histogram z próby dobrze przybliża gęstość generowanego rozkładu,
3. punkty na wykresie kwantylowej układają się w jedną linię.

Wszystkie przedstawione metody pokazywały zgodność z rozkładami teoretycznymi. Niezależnie czy korzystaliśmy z najpopularniejszych rozkładów prawdopodobieństwa, czy stworzonych samodzielnie, żadna z metod nie wykazała żadnych znaczących defektów.

Dodatkowo pokazaliśmy, że algorytm w metodzie biegunowej jest znacznie szybszy niż algorytm dla metody B-M. Ma to duże znaczenie, ponieważ obie metody generują próbę z rozkładu normalnego. Rozkład ten jest bardzo ważny, dlatego każdy sposób przyśpieszenia generowania tego rozkładu jest na wagę złota, szczególnie, że w tym przypadku różnica między czasem działania obu algorytmów jest znacząca.

Bibliografia

- [1] <https://eportal.pwr.edu.pl/mod/resource/view.php?id=329296>
- [2] https://en.wikipedia.org/wiki/Linear_congruential_generator
- [3] <https://youtu.be/oJYFlu2oJQE?t=19>
- [4] <https://youtu.be/NFmbgbyj5M0?t=32>
- [5] <https://youtu.be/NFmbgbyj5M0?t=1323>
- [6] <https://youtu.be/SpPS0CnhvrE?t=37>
- [7] <https://youtu.be/SpPS0CnhvrE?t=4081>
- [8] <https://youtu.be/7lOH982wrwo?t=50>
- [9] <https://youtu.be/7lOH982wrwo?t=1896>
- [10] <https://youtu.be/7lOH982wrwo?t=3349>