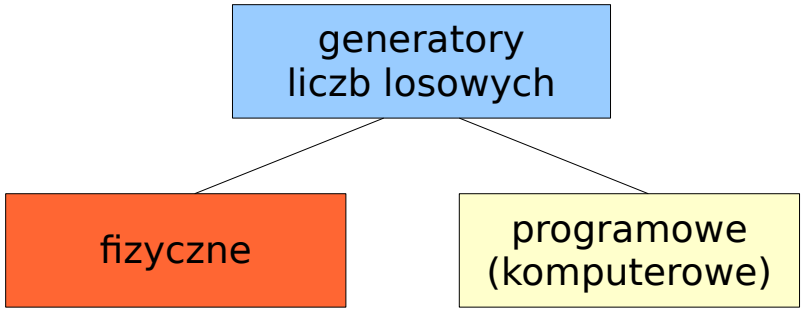
Generatory liczb pseudolosowych

Plan wykładu:

- liniowe generatory o rozkładzie równomiernym
- generatory o dowolnym rozkładzie prawdopodobieństwa
    - metoda odwracania dystrybuanty
    - metoda eliminacji
    - rozkład dyskretny
- generatory o rozkładach wielowymiarowych
    - rozkład równomierny na sferze i kuli w R^M
    - dwuwymiarowy rozkład normalny
- testowanie generatorów
-
    - testy zgodności z rozkładem: χ^2

Generatory o rozkładzie równomiernym



Najprostsze generatory liczb losowych to generatory fizyczne wykorzystujące

- szumy układów elektronicznych
- promieniotwórczość

- zalety: dostajemy ciągi liczb losowych (niezależne, nieskorelowane)

- wady: wymagana ciągła kalibracja (testowanie parametrów),
  kłopoty techniczne z obsługą,
  brak powtarzalności serii

Własności generatorów komputerowych

- łatwość obsługi
- możliwość generowania dowolnego rozkładu
- dowolna liczba wymiarów
- powtarzalność ciągów generowanych liczb

Jak pracują komputerowe generatory liczby?

1) Tworzony jest ciąg liczb nieujemnych (naturalnych)

$$X_0, X_1, X_2, \ldots, X_n$$

o **rozkładzie równomiernym** według wybranego algorytmu.
Liczby ograniczone są od góry przez reprezentację,
np. dla k=32 bitowej reprezentacji liczby całkowitej bez znaku, kres górny

$$m = 2^{32}$$

$$\underbrace{X_0, \ldots, X_{\nu-1}}_{T_a} \underbrace{X_\nu, X_{\nu+1}, \ldots, X_{\nu+P-1}}_{T_o}, X_{\nu+P}, \ldots$$

T_a – okres aperiodyczności ciągu
T_0 – okres ciągu

2) Aby uzyskać liczby „rzeczywiste” dokonujemy przekształcenia

$$U = \frac{X}{m} \Rightarrow U_i \in (0, 1]$$

3) Dokonujemy kolejnej transformacji ciągu aby uzyskać ciąg o zadanym rozkładzie prawdopodobieństwa (normalny, wielomianowy, etc.)

Generatory liniowe

Generatory liniowe tworzą ciąg liczb według schematu:

$$X_{n+1} = (a_1 X_n + a_2 X_{n-1} + \ldots + a_k X_{n-k+1} + c) \; mod \; m$$

gdzie: $a_1, a_2, \ldots, a_k$, c, m – parametry generatora (ustalone liczby)

Operację

$$r = (a \; mod \; n), \quad a, n, r \in Z$$

nazywamy dzieleniem modulo a jej wynikiem jest reszta z dzielenia liczb całkowitych a i n.

Lub inaczej: r **jest kongruentne do a modulo n jeśli n jest dzielnikiem a-r**.

$$a \equiv r \; mod \; n \Rightarrow r = a - \left\lfloor \frac{a}{n} \right\rfloor n$$

Generatory wykorzystujące operację dzielenia modulo to generatory **kongruentne** lub **kongruencyjne**.

Przykład

19 mod 6 =**1**	15 mod 6 =3
18 mod 6 =0	14 mod 6 =2
17 mod 6 =5	13 mod 6 =**1**
16 mod 6 =4	12 mod 6 =0

$$X_{n+1} = (a_1 X_n + a_2 X_{n-1} + \ldots + a_k X_{n-k+1} + c) \; mod \; m$$

Aby wygenerować ciąg liczb pseudolosowych należy zdefiniować jego parametry.

- liczby $X_0, X_1, X_2, \ldots, X_k$

  nazywamy ziarnem generatora (seed)

Dla bardziej rozbudowanych generatorów liczby te otrzymujemy z innego generatora lub np. używając zegara systemowego (X_0).

- najprostszy generator liniowy ma dwie odmiany
  - generator multiplikatywny $c = 0$
  - generator mieszany $c \neq 0$
- maksymalny okres generatora liniowego to (m-1)

- najprostszy generator multiplikatywny

$$X_{i+1} = aX_{i-1} \ mod \ m$$

$$k_i = \left\lfloor \frac{aX_{i-1}}{m} \right\rfloor, \quad i \geq 1$$

$$X_1 = aX_0 - mk_1$$
$$X_2 = a^2X_0 - mk_2 - mk_1a$$
$$X_3 = a^3X_0 - mk_3 - mk_2a - mk_1a^2$$
$$\ldots\ldots\ldots\ldots$$
$$X_n = a^nX_0 - m(k_n + k_{n-1}a + \ldots + k_1a^{n-1})$$

ostatnie równanie można zapisać w postaci

$$X_n = a^nX_0 \ mod \ m$$

- wybór X_0 determinuje wszystkie liczby w generowanym ciągu (a i m są ustalone)
- działa w sposób całkowicie **deterministyczny** !!!
- całkowity brak losowości
- taki generator jest **NIEPRZYDATNY**

przykład: Wpływ parametrów generatora na wyznaczane ciągi liczb

$$X_i = aX_{i-1} \ mod \ 11, \quad X_0 = 1$$

					X_i					
	a=1	2	3	4	5	6	7	8	9	10
i=0	1	**1**	1	1	1	**1**	**1**	**1**	1	1
1	1	**2**	3	4	5	**6**	**7**	**8**	9	10
2		**4**	9	5	3	**3**	**5**	**9**	4	1
3		**8**	5	9	4	**7**	**2**	**6**	3	
4		**5**	4	3	9	**9**	**3**	**4**	5	
5		**10**	1	1	1	**10**	**10**	**10**	1	
6		**9**				**5**	**4**	**3**		
7		**7**				**8**	**6**	**2**		
8		**3**				**4**	**9**	**5**		
9		**6**				**2**	**8**	**7**		
10		**1**				**1**	**1**	**1**		

- okres generatora zależy od jego parametrów
- źle dobrane parametry ograniczają okres generatora

- okres generatora multiplikatywnego

$$T = min\{i : X_i = X_0, i > 0\}$$

- maksymalny okres generatora multiplikatywnego uzyskujemy dla

$$a^{(m-1)/p} \neq 1 \ mod \ m$$

Gdy m jest liczbą pierwszą a p jest czynnikiem pierwszym liczby (m-1).

przykład: wykorzystujemy **liczby Mersenne'a**
(które dość często są liczbami pierwszymi)

$$m = 2^p - 1 \quad \Rightarrow \quad p = 31 \Rightarrow m = 2^{31} - 1, \quad a = 7^5$$

- okres generatora

$$T = 2^{31} - 2$$

- liczby

$$U = \{U_i, 1 \leq i \leq T\}$$

występują dokładnie 1 raz w pojedynczym okresie generatora

- średnia odległość pomiędzy najbliższymi sąsiadami

$$\frac{1}{2^{31} - 1} = 4.657 \times 10^{-10}$$

Rozkład przestrzenny ciągu

- wadą generatorów multiplikatywnych jest nierównomierne pokrycie d-wymiarowej kostki (I^d)
- generowane liczby lokalizują się na hiperpłaszczyznach, których położenie uzależnione jest od parametrów generatora
- jak badamy pojawienie się zależności między kolejnymi elementami ciągu?
  tworzymy ciągi wektorów d-wymiarowych z wylosowanych liczb

  - wektory utworzone z częściowo przekrywających się ciągów liczb

$$(U_1, U_2, \ldots, U_d), (U_2, U_3, \ldots, U_{d+1}), \ldots$$

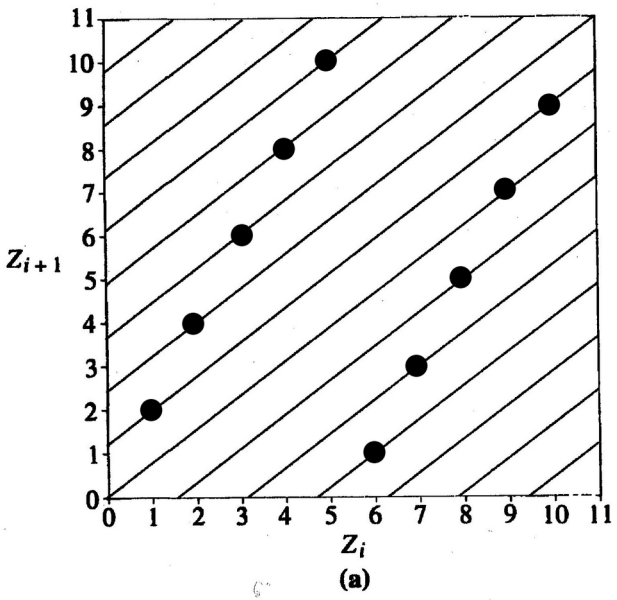
  - wektory utworzone z ciągów rozdzielnych

$$(U_1, U_2, \ldots, U_d), (U_{d+1}, U_{d+2}, \ldots, U_{2d}), \ldots$$

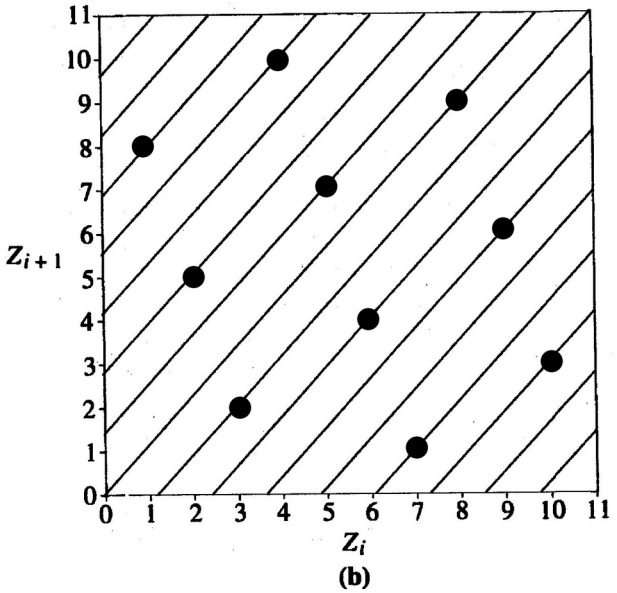
przykład: generator multiplikatywny – korelacja elementów ciągu – test graficzny

$$X_i = aX_{i-1} \ mod \ 11$$

$X_0 = 1, \quad a = 2$



$X_0 = 1, \quad a = 8$

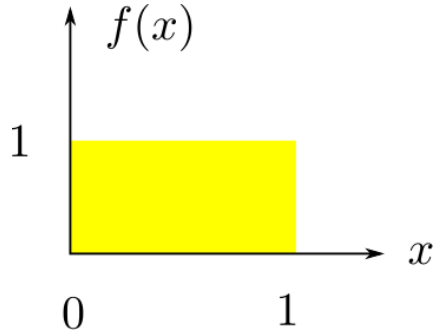


wnioski:

- część przestrzeni nie jest pokryta punktami – zostaje wyłączona z obliczeń
- ewidentny brak losowości
- zmiana parametrów nie eliminuje wady

Podstawowe parametry statystyczne generatora o rozkładzie równomiernym U(0,1)

- zakładamy, że generator dostarcza liczb losowych o rozkładzie jednorodnym w zakresie x=(0,1), rozkład definiuje funkcja gęstości prawdopodobieństwa f(x)



$$f(x) = 1, \quad x \in (0,1)$$

- jeśli generowany ciąg liczb jest niezależny to **wartość oczekiwana zmiennej losowej** powinna wynosić

$$\mu = \int_0^1 \underbrace{f(x)}_{=1} x dx = \frac{x^2}{2}\Big|_0^1 = \frac{1}{2}$$

jej estymatorem jest średnia arytmetyczna

$$\bar{\mu} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- **wariancja rozkładu** zdefiniowana jest jako drugi moment centralny

$$\sigma^2 = \int_0^1 (x-\mu)^2 dx = \frac{1}{12}$$

$$\bar{\sigma} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{\mu})^2$$

- jeśli parametry statystyczne generatora (ciągu generowanych przez niego liczb) odbiegają od powyższych wartości to jest on nieprzydatny (lub warunkowo przydatny)

- ponadto współczynniki autokorelacji elementów ciągu powinny wynosić 0

Funkcja autokorelacji

- opisuje zależność elementów ciągu od wyrazów poprzednich
- definicja

$$R_r = \frac{E[(X_t - \mu)(X_s - \mu)]}{\sigma^2}$$

oraz wzór dla ciągu skończonego (r=s-t to przesunięcie elementów)

$$\overline{R}_r = \frac{1}{(N-r)\sigma^2} \sum_{i=1}^{N-r} (X_i - \mu)(X_{i+r} - \mu)$$

- inaczej: opisuje związek pomiędzy elementami dwóch szeregów – danego i przesuniętego o r

- dla rozkładu jednorodnego R~0 oznacza brak korelacji pomiędzy elementami ciągu, czyli rozkład liczb pseudolosowych jest stochastyczny (brak korelacji)

- w praktyce najsilniejsza jest korelacja między kilkoma kolejnymi liczbami pseudolooswymi, wyznacza się funkcję autokorelacji dla

$$r = 1, 2, 3, 4, 5, 6$$

przykład: generatory multiplikatywne o dobrych własnościach statystycznych

$$X_i = (1176X_{i-1} + 1476X_{i-2} + 1776X_{i-3}) \ mod \ (2^{32} - 5)$$
$$X_i = 2^{13}(X_{i-1} + X_{i-2} + X_{i-3}) \ mod \ (2^{32} - 5)$$
$$X_i = (1995X_{i-1} + 1998X_{i-2} + 2001X_{i-3}) \ mod \ (2^{35} - 849)$$
$$X_i = 2^{19}(X_{i-1} + X_{i-2} + X_{i-3}) \ mod \ (2^{32} - 1629)$$

- generatory te posiadają maksymalny okres

$$T = m - 1$$

Generatory (bitowe) na rejestrach przesuwnych

- tworzymy ciąg bitów b_i otrzymywanych rekurencyjnie

$$b_i = (a_1 b_{i-1} + \ldots + a_k b_{i-k}) \ mod \ 2 \qquad i = k+1, k+2, \ldots$$

$a_1, a_2, \ldots, a_k \in \{0,1\}$ - parametry generatora są bitami

$b_1, b_2, \ldots, b_k \in \{0,1\}$ - ciąg inicjujący

- do tworzenia ciągu bitów możemy wykorzystać operację **XOR**

a	b	a xor b
0	0	0
0	1	1
1	0	1
1	1	0

$$a \ xor \ b = (a+b) \ mod \ 2$$

- jeśli założymy, że parametry generatora są jedynkami

$$a_{j1} = a_{j2} = \ldots = a_{jk} = 1$$

kolejny element ciągu tworzymy stosując operację XOR sekwencyjnie

$$b_i = b_{i-j_1} \ xor \ b_{i-j_2} \ xor \ \ldots \ xor \ b_{i-j_k}$$

- okres generatora w tej postaci to $T = 2^k - 1$

- w praktyce używa się tylko bitów do generacji kolejnego (większa wydajność)

$$b_i = b_{i-p} \ xor \ b_{i-q}, \quad p > q, \quad p, q \in N$$

Generator bitowy Tauswortha

- wykorzystujemy ciąg bitów do obliczenia liczby pseudolosowej z przedziału (0,1]

$$U_i = \sum_{j=1}^{L} 2^{-j} b_{is+j} = 0.b_{is+1} \dots b_{is+L}, \qquad i = 0, 1, 2, \dots$$

gdzie: s jest ustaloną całkowitą liczbą nieujemną

- jeśli s<L to do utworzenia U_i oraz U_{i+1} wykorzystywane są elementy tego samego podciągu
- jeśli s=L to U_i oraz U_{i+1} są tworzone z rozłącznych fragmentów ciągu globalnego

- ciąg bitów łatwo generuje się przy użyciu rejestrów przesuwnych oraz bramek logicznych (**XOR**) – łatwa implementacja w języku C

- modyfikacja tego typu generatora to generator **Mersenne Twister** (1998)

- okres generatora jest ogromy $T = 2^{19937} - 1$

- generator potrafi wypełnić liczbami pseudolosowymi w sposób jednorodny 623-wymiarową kostkę

$$U^d(0,1), \quad d = 623$$

Generatory o dowolnym rozkładzie prawdopodobieństwa

- generator o rozkładzie jednorodnym U(0,1) to podstawowy typ generatora, który wykorzystywany jest do konstrukcji generatorów o innych rozkładach

- do konstruowania generatorów o dowolnym rozkładzie stosujemy

  - metodę odwracania dystrybuanty (najszybsza, ale nie zawsze stosowalna)
  - metodę łańcuchów Markowa (szybka i skuteczna, ale występują korelacje w ciągu)
  - metodę eliminacji von Neumanna (najwolniejsza, ale zawsze działa)

Metoda odwracania dystrybuanty

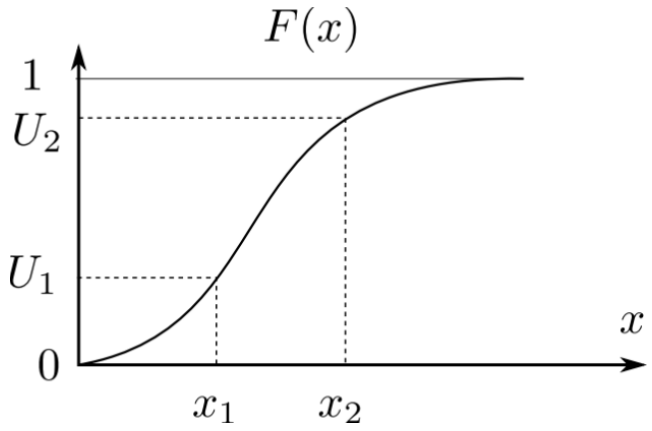
- rozkład prawdopodobieństwa w sposób jednozanczny określają dwie funkcje **dystybuanta rozkładu** i **funkcja geśtości prawdopodobieństwa**

- dystrybuanta jest funkcją niemalejącą i prawostronnie ciągłą

$$F : R \to R$$

$$\lim_{x \to -\infty} F(x) = 0$$

$$\lim_{x \to \infty} F(x) = 1$$



- funkcja gęstości prawdopodobieństwa jest nieujemna i unormowana

$$f(x) \geq 0, \qquad \int_{-\infty}^{\infty} f(x)dx = 1$$

- dystrybuanta i fgp rozkładu są ze sobą ściśle związane (są miarą prawdopodobieństwa)

$$F(x) = \int_{-\infty}^{x} f(y)dy$$

- znajdźmy funkcję odwrtoną dystrybuanty, co wówczas uzyskamy?

$$U = F(x) \to x = F^{-1}(U), \qquad U \in [0,1], \qquad x \in (-\infty, \infty)$$

- wstawiając jako argument do funkcji odwrotnej liczbę losową o rozkładzie jednorodnym U(0,1) dokonujemy jej transformacji uzyskując liczbę losową o rozkładzie zdefiniowanym przez dystrybuantę

przykład: metoda odwracania dystrybuanty - **rozkład jednomianowy**

- funkcja gęstości prawdopodobieństwa to

$$f(x) = (n+1)x^n$$

a pozostałe parametry rozkładu przyjmijmy w postaci

$$\mathrm{x} \in [0,1], \qquad n = 1, 2, 3, \ldots$$

- określamy funkcję opisującą dystrybuantę

$$F(x) = (n+1)\int_0^x (x')^n dx' = (n+1)\frac{x^{n+1}}{n+1} = U$$

i uzależniamy x od U

$$x = U^{\frac{1}{n+1}}, \qquad x \in (0,1)$$

- generator o rozkładzie jednomianowym

przykład: metoda odwracania dystrybuanty - **rozkład eksponencjalny**

- funkcja gęstości prawdopodobieństwa rozkładu eksponencjalnego

$$f(x) = e^{-x}, \quad x \in [0, \infty)$$

- dystrybuanta

$$F(x) = \int_0^x e^{-x'} dx'$$

$$F(x) = 1 - e^{-x} = U, \qquad U \in (0, 1)$$

$$e^{-x} = 1 - U$$

$$F^{-1}(x) = x = -\ln(1 - U)$$

$$\boxed{x = -ln(1 - U)}$$

- generator o rozkładzie eksponencjalnym

przykład: metoda odwracania dystrybuanty - **rozkład normalny N(0,1)**

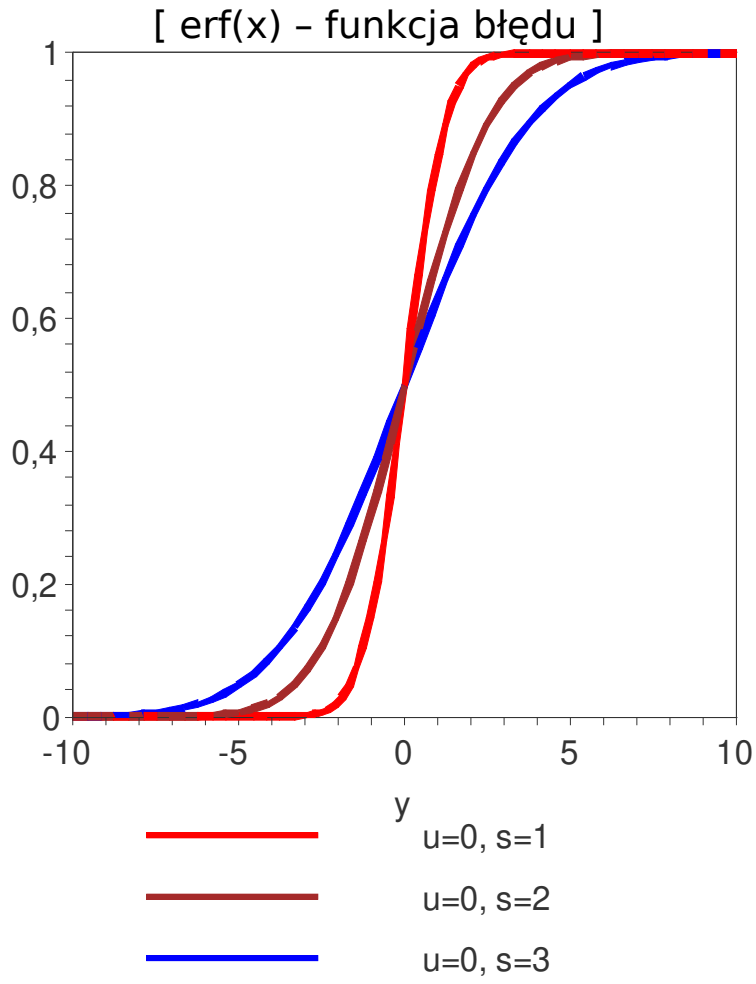
- fgp to funkcja Gaussa

$$f(x) = e^{-x^2}$$

- dystrybuanta jest funkcją błędu

$$F(x) = \int_{-\infty}^{x} e^{-x'^2} dx' = erf(x)$$

- szukanie funkcji odwrotnej erf(x) jest kosztowne, dlatego stosuje się metodę Boxa-Mullera.



Metoda Boxa-Mullera dla rozkładu normalnego

- definiujemy fgp w **2D** jako złożenie dwóch funkcji gaussowskich

$$f(x,y) = f(x) \cdot f(y) = e^{-\frac{x^2+y^2}{2}}, \qquad x, y \in (-\infty, \infty)$$

- docelowo chcemy policzyć prawdopodobieństwo

$$p(x,y) = f(x,y)dxdy$$

(czyli że wylosowana liczba znajdzie się w obszarze dxdy)

- dokonujemy transformacji: współrzędne kartezjańskie → współrzędne biegunowe

$$x = r\cos(\theta) \qquad r \in [0, \infty)$$

$$y = r\sin(\theta) \qquad \theta \in [0, 2\pi]$$

$$\Rightarrow \quad r^2 = x^2 + y^2$$

- prawdopodobieństwo określmy przy użyciu nowych zmienych – stosujemy prawo przenoszenia prawdopodobieństwa

$$p = f(x,y)dxdy = f(r,\theta)\, r\, dr\, d\theta$$

$$p(r,\theta) = r \cdot e^{-r^2/2} dr d\theta$$

- dokonaliśmy **separacji zmiennych** (r,θ)

- wprowadzamy jeszcze raz nową zmienną (ułatwi nam to całkowanie)

$$z = \frac{r^2}{2} \rightarrow dz = r dr, \qquad z \in [0, \infty)$$

$$p(z, \theta) = e^{-z} dz d\theta = f(z) dz \cdot 1 \cdot d\theta$$

- dostaliśmy **rozkład wykładniczy**, który wiemy jak wygenerować

$$f(z) = e^{-z} \quad \rightarrow \quad z = -\ln(1 - U_1), \quad U_1 \in (0, 1)$$

następnie podstawiamy: $r = \sqrt{2z} = \sqrt{-2ln(1-U)}$

- kąt θ ma **rozkład jednorodny**, więc używamy generatora o rozkładzie jednorodnym

$$\theta = U_2 \cdot 2\pi, \quad U_2 \in (0, 1)$$

- dla pary (U_1,U_2) dostajemy parę liczb losowych (x,y) z rozkładu N(0,1)

$$X = r\cos(\theta) = \sqrt{-2\ln(1-U_1)}\cos(2\pi U_2)$$
$$Y = r\sin(\theta) = \sqrt{-2\ln(1-U_1)}\sin(2\pi U_2)$$

- jeśli chcemy zmienić parametry rozkładu normalnego, to dokonujemy transformacji liniowej

$$X = x \cdot \sigma + \mu \qquad x \in N(0,1) \rightarrow X \in N(\mu, \sigma)$$

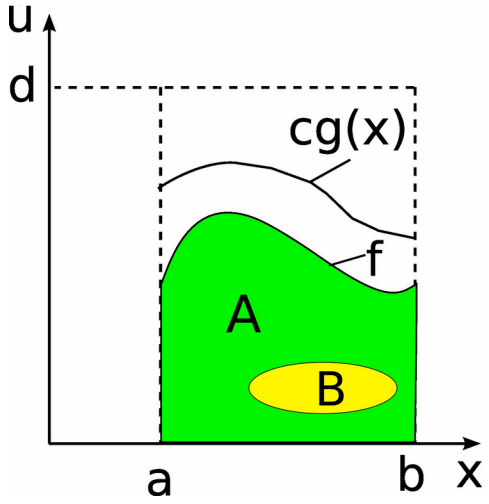
Metoda eliminacji von Neumann'a

- chcemy wygenerować ciąg zmiennych losowych o gęstości prawdopodobieństwa f w przedziale [a,b]
- wartość f jest w przedziale [a,b] ograniczona od góry przez stałą d

Sposób otrzymania ciągu zmiennych losowych o rozkładzie f(x) jest następujący:

- losujemy dwie zmienne o rozkładzie równomiernym

$$U_1 \in [a,b] \qquad \mathrm{U}_2 \in [0,d]$$



- sprawdzamy warunek

$$U_2 \leq f(U_1) \Rightarrow X = U_1$$

  - gdy powyższy warunek nie jest spełniony wówczas odrzucamy parę U_1, U_2, jeśli warunek jest spełniony to zachowujemy nowy element $X_i=U_1$
  - losowanie wg tego schematu wykonujemy, aż do uzyskania odpowiednio licznego ciągu tj. zaakceptowanych liczb $X_i=U_1$

- wygenerowany ciąg liczb $\{X_1, X_2, X_3,...\}$ ma rozkład zdefiniowany przez funkcję gęstości prawdopodobieństwa f

Generowanie ciągu liczb pseudolosowych o zadanym rozkładzie algorytmem Metropolisa

- modyfikujemy metodę eliminacji: w standardowej postaci jest ona mało wydajna, bo nierzadko odrzucamy większość wyników, dzięki **algorytmowi Metropolisa** nie odrzucamy żadnego wyniku

- akceptację nowego położenia (nowej liczby w ciągu) dokonujemy zgodnie z formułą

$$h(x_{i-1}, x_i) = \min\left(1, \frac{f(x_i)}{f(x_{i-1})}\right)$$

- jeśli $f(x_i) > f(x_{i-1})$ to nowe położenie akceptujemy zawsze

- w przeciwnym wypadku akceptacja następuje z prawdopodobieństwem $f(x_i)/f(x_{i-1})$

- jeśli nie akceptujemy nowego punktu to zatwierdzamy stary (każdy krok generuje nowy element ciągu)

- nowe, proponowane położenie losujemy w taki sposób, aby zachowana była symetria prawdopodobieństwa przejścia między dwoma stanami losowymi (tzw. **warunek detailed balance**)

$$p(x_i \rightarrow x_{i+1}) = p(x_{i+1} \rightarrow x_i)$$

przykład: zastosowanie algorytmu Metropolisa do generowania liczb pseudolosowych

- szukamy ciągu liczb pseudolosowych dla fgp

$$f(x) = \frac{1}{2}x^2 e^{-x}$$

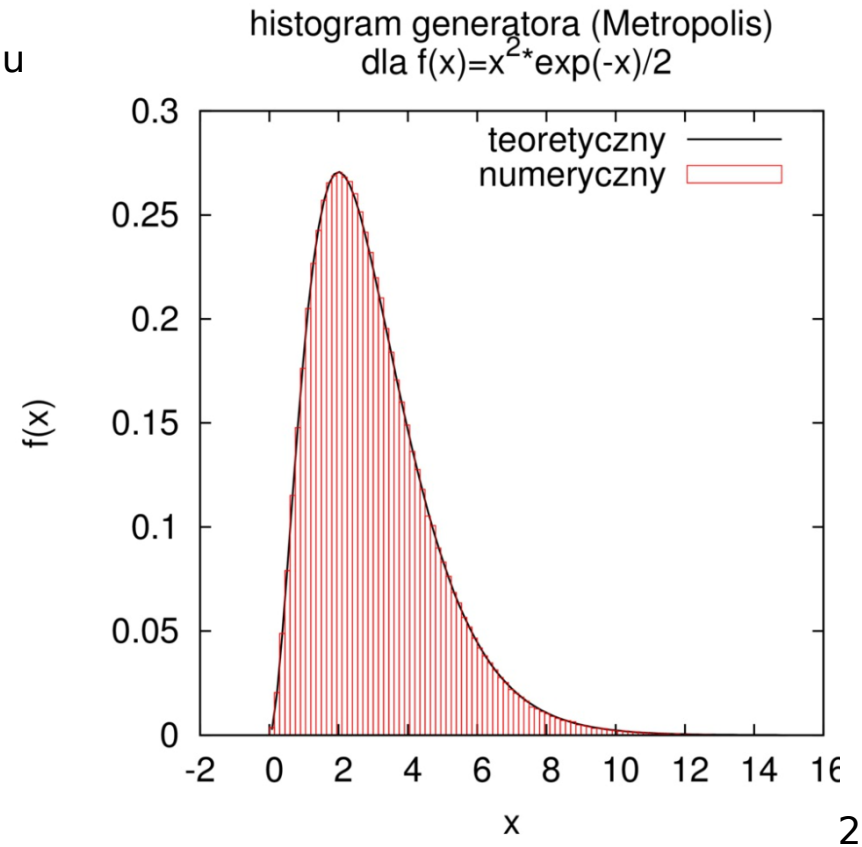
- nowego proponowanego stanu losowego (liczby) szukamy w symetrycznym względem punktu x_i przedziale o szerokości 2

$$x_{new} = x_i + (2 \cdot U(0,1) - 1)$$

- następnie sprawdzamy warunki akceptacji nowego stanu

$$h(x_{i-1}, x_i) = \min\left(1, \frac{f(x_i)}{f(x_{i-1})}\right)$$

$$x_{i+1} = \begin{cases} x_i & \Longleftrightarrow \quad x_{new} < 0 \\ x_i & \Longleftrightarrow \quad x_{new} > 0 \quad \wedge u \in U(0,1) > h \\ x_{new} & \Longleftrightarrow \quad x_{new} > 0 \quad \wedge u \in U(0,1) < h \end{cases}$$



Metoda odwracania dystrybuanty dla rozkładu dyskretnego

- w rozkładzie dyskretnym mamy określone prawdopodobieństwo wylosowania danej liczby (rozkład może nie być równomierny)

$$P\{K = k\} = p_k, \quad k = 0, 1, 2, \ldots, M \qquad \sum_{k=0}^{M} p_k = 1$$

- algorytm generowania ciągu zmiennych o rozkładzie dyskretnym:

1) przyjmujemy wartości początkowe zmiennych: K=0, S=p_0

2) losujemy zmienną U o rozkładzie równomiernym z przedziału [0,1]

3) sprawdzamy warunek: U<S

   - jeśli jest prawdziwy to zachowujemy X=k i dokonujemy kolejnego losowania (pkt. 1)
   - jeśli nie jest spełniony to iteracyjnie obliczamy (zwiększamy S)

$$K = K + 1, \quad S = S + p_k$$

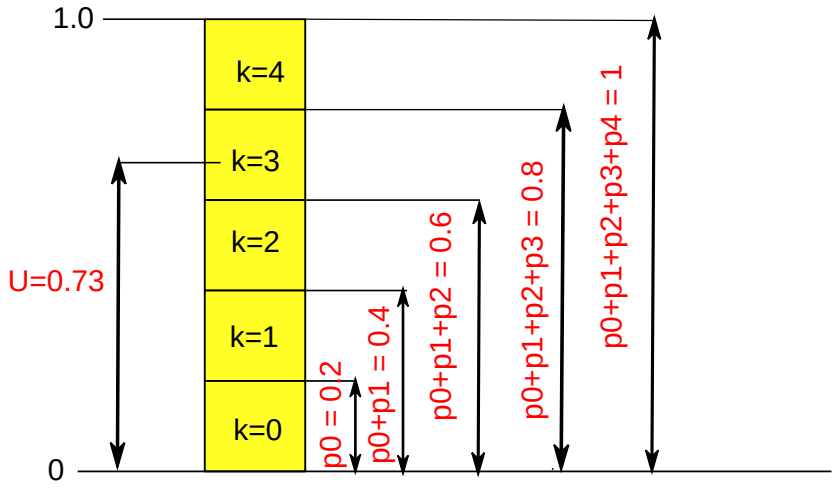
   dopóki warunek jest spełniony, gdy przestanie być prawdziwy zachowujemy ostatnią wartość K dołączając ją do ciągu liczb

4) Kroki 1-3 wykonujemy aż do uzyskania odpowiednio licznego ciągu

{K_1, K_2, K_3,...}

przykład: odwracanie dystrybuanty dla rozkładu dyskretnego

$$k = 0, 1, 2, 3, 4 \qquad \rightarrow \qquad p_0 = p_1 = p_2 = p_3 = p_4 = 0.2$$



$U = 0.73$

$K = 0, \quad S = p_0, \quad U > S\,(?) \quad \text{TAK}$

$K = 1, \quad S = p_0 + p_1, \quad U > S\,(?) \quad \text{TAK}$

$K = 2, \quad S = p_0 + p_1 + p_2, \quad U > S\,(?) \quad \text{TAK}$

$K = 3, \quad S = p_0 + p_1 + p_2 + p_3, \quad U > S\,(?) \quad \text{NIE}$

$(0.73 < 0.8)$

Generatory o rozkładach wielowymiarowych

- zadanie można sformułować następująco:

  należy wygenerować ciąg wielowymiarowych zmiennych (wektorów) losowych

$$\boldsymbol{X} = (X_1, X_2, \ldots, X_k)$$

  których rozkład prawdopodobieństwa ma gęstość

$$f(x_1, x_2, \ldots, x_k)$$

- do generacji takiego ciągu można stosować metodę eliminacji,
  ale w jej najprostszej postaci pojawiają się problemy z wydajnością
  (duża liczba odrzuconych wyników)

Przykład: Określić prawdopodobieństwo akceptacji wielowymiarowej zmiennej losowej o rozkładzie równomiernym na kuli jednostkowej ($K_k(0,1)$).

- zastosujmy prosty algorytm eliminacji
  - losujemy m zmiennych niezależnych o rozkładzie równomiernym w (-1,1)
    i konstruujemy zmienną wielowymiarową

$$\boldsymbol{U} = (U_1, U_2, \ldots, U_k)$$

  - zmienną akceptujemy jeśli

$$\|\boldsymbol{U}\|_2 \leq 1$$

    w przeciwnym wypadku, odrzucamy ją bo leży poza kulą.

- gdzie leży problem w wydajności metody eliminacji?

- prawdopodobieństwo akceptacji zmiennej jest równe ilorazowi objętości kuli i opisanej na niej kostki $[-1,1]^k$

$$V_{kula}(r) = \frac{\pi^{\frac{m}{2}}}{\Gamma\left(\frac{m}{2}+1\right)} r^m \qquad V_{kostka}(r) = 2^m \qquad \Rightarrow \qquad p_m = \frac{\pi^{\frac{m}{2}}}{2^m \Gamma\left(\frac{m}{2}+1\right)}$$

- średnia liczba wylosowanych punktów N_m potrzebnych do realizacji jednej zmiennej wynosi

$$N_m = \frac{1}{p_m}$$

m	p_k	N_k
2	7.854×10^{-1}	1.27
5	1.645×10^{-1}	6.08
10	2.490×10^{-3}	4.015×10^{2}
20	2.461×10^{-8}	4.063×10^{7}
50	1.537×10^{-28}	6.507×10^{27}

← 1 akceptacja na 400 prób !!!

- modyfikacją usprawniającą powyższy algorytm jest podział kostki na rozłączne podobszary i przeprowadzenia losowania w każdym z nich z osobna

- generalnie dla większej liczby wymiarów, metoda eliminacji jest nieprzydatna

Jednorodny rozkład gęstości prawdopodobieństwa w kole $K^2(0,1)$

- w pewnych (**nielicznych**) przypadkach do generowania ciągu liczb o zadanym rozkładzie wielowymiarowym możemy użyć metodę odwracania dystrybuanty
- zastosujmy ją do znalezienia rozkładu jednorodnego w kole o promieniu R=1

- ogólny wzór na powierzchnię

$$S = \int_S dS$$

- element powierzchnii we wsp. radialnych

$$dS = r d\varphi \cdot dr$$

- potraktujmy funkcję podcałkową jako fgp

$$S = \int_0^{2\pi} d\varphi \int_0^1 r dr = S \cdot \int_0^{2\pi} f_\varphi(\varphi) d\varphi \cdot \int_0^1 f_r(r) dr$$

$$f_\varphi(\varphi) = \frac{1}{2\pi}$$

$$f_r(r) = 2r$$

ogólnie dla k-wymiarów:

$$f_r^{(k)}(r) = k \cdot r^{k-1}$$

- losując dwie liczby losowe z rozkładów f_φ i f_r mamy gwarancję, że znajdują się one wewnątrz koła o promieniu 1

Rozkład równomierny na k-wymiarowej sferze $S^k(0,1)$ i w kuli $K^k(0,1)$

- jeżeli k-wymiarowa zmienna losowa

$$\boldsymbol{X} = (X_1, X_2, \ldots, X_k)$$

położona jest na sferze o promieniu R=1 to spełniony jest warunek

$$S_k = \left\{ (x_1, \ldots, x_k) : \sum_{i=1}^{k} x_i^2 = 1 \right\}$$

- k-wymiarowa zmienna X ma rozkład **sferycznie konturowany** jeżeli jej funkcja gęstości prawdopodobieństwa od długości wektora będącego jej argumentem

$$f(x_1, x_2, \ldots, x_k) = f(\vec{x}) = f(\|\vec{x}\|) = f\left(\sum_{i=1}^{k} x_i^2 \right)$$

- dysponując zmienną o rozkładzie sferycznie konturowanym wygenerujemy zmienną o rozkładzie równomiernym:
  - na sferze poprzez znormalizowanie długości wektora x do promienia R
  - wewnątrz kuli poprzez przeskalowanie dłogości wektora wg rozkładu f_r

Rozkład sferycznie konturowany – rozkład normalny (Gaussa)

- k-wymiarowy rozkład normalny opisuje gęstość

$$f_X(x_1,\ldots,x_k) = \frac{1}{(2\pi)^{k/2}} \exp\left(-\frac{1}{2}\|\vec{x}\|^2\right)$$

- algorytm generowania rozkładu równomiernego na sferze w k-wymiarach:
  - generujemy m-wymiarową zmienną losową **X** o rozkładzie normalnym (generator Boxa-Mullera)
  - dokonujemy skalowania, umieszczając punkty (k-wymiarów) na sferze

$$\vec{X} = \frac{\vec{x}}{\|\vec{x}\|}$$

- rozkład jednorodny w kuli k-wymiarowej
  - losujemy długość wektora leżącego wewnątrz kuli z rozkładu

$$R \quad \leftarrow \quad f_r(r) = k \cdot r^{k-1}, \quad 0 \le r \le 1$$

  - dokonujemy ponownego skalowania długości wektora

$$\mathrm{Z}=(\mathrm{RX}_1, RX_2, \ldots, RX_k)$$

Testowanie generatorów liczb pseudolosowych

- wszystkie generatory o dowolnym rozkładzie bazują na wykorzystaniu ciągów liczb pseudolosowych o rozkładzie równomiernym więc istotne jest badanie generatorów liczb o takim właśnie rozkładzie

- badanie generatorów o dowolnym rozkładzie można ograniczyć do sprawdzenia czy generowany rozkład jest zgodny z rozkładem docelowym

- testowanie generatora jest procesem złożonym:

  - dla ustalonej liczby n, generujemy n kolejnych liczb startując od losowo wybranej liczby początkowej

  - obliczamy wartość statystyki testowej (T) (np. test χ^2)

  - obliczamy F(T) czyli dystrybuantę statystyki T, gdy weryfikowana hipoteza jest prawdziwa

    kroki 1-3 powtarzamy N-krotnie obliczając statystyki: $T_1, T_2, \ldots, T_N$.

- jeśli weryfikowana hipoteza jest prawdziwa to

$$F(T_1), F(T_2), F(T_3), \ldots, F(T_N)$$

jest ciągiem zmiennych niezależnych o rozkładzie równomiernym, testowanie generatora kończy się sprawdzeniem tej hipotezy

Testy zgodności z zadanym rozkładem – test chi-kwadrat

- badamy w nim hipotezę że generowana zmienna losowa X ma rozkład prawdopodobieństwa o dystrybuancie F

jeżeli

$$F(a) = 0 \quad F(b) = 1$$

to możemy dokonać następującego podziału zbioru wartości zmiennej X

$$a < a_1 < a_2 < \ldots < a_k = b \quad \Rightarrow \quad p_i = P\{a_{i-1} < X \leq a_i\}, \quad i = 1, 2, \ldots$$

Generujemy ciąg n liczb

$$X_1, X_2, \ldots, X_n$$

Sprawdzamy ile z nich spełnia warunek

$$a_{i-1} < X \leq a_i$$

ich liczbę oznaczamy $\mathbf{n_i}$. Statystyką testu jest

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - np_i)^2}{np_i}$$

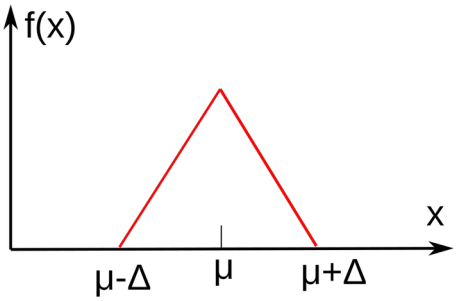
dla dużego n statystyka ta ma rozkład χ^2 o (k-1) stopniach swobody.

- uzyskaną wartość porównujemy z wartością graniczną dla rozkładu chi-kwadrat (np. korzystając z tabel statystycznych), jeśli jest mniejsza od wartości granicznej to hipotezy nie odrzucamy

przykład: test generatora o trójkątnym rozkładzie fgp

$$f(x;\mu,\Delta) = -\frac{|x-\mu|}{\Delta^2} + \frac{1}{\Delta}$$

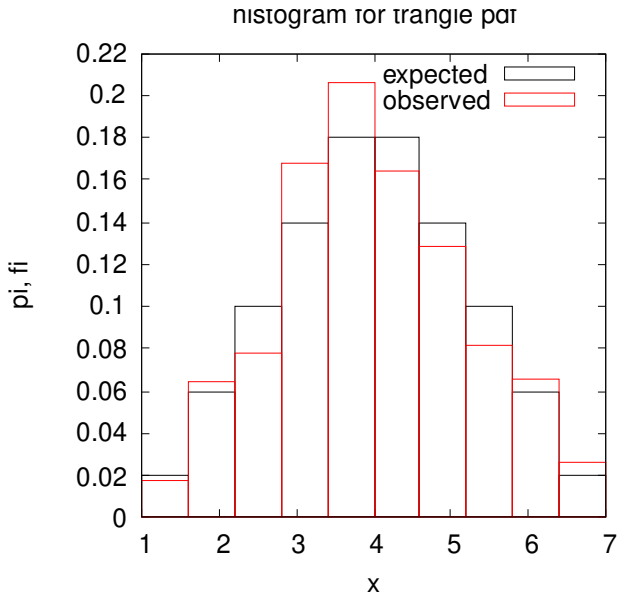
mamy 2 parametry
więc 2 stopnie swobody
mniej:
(k-1-2)



$$F(a) = P(x<a) = \int_{\mu-\Delta}^{a} f(x:\mu,\Delta)dx = \begin{cases} -\frac{1}{\Delta^2}\left(-\frac{x^2}{2}+\mu x\right)+\frac{x}{\Delta}, & x \le \mu \\ -\frac{1}{\Delta^2}\left(\frac{x^2}{2}-\mu x+\mu^2\right)+\frac{x}{\Delta}, & x > \mu \end{cases}$$

generowanie zmiennej o rozkładzie trójkatnym

$$x = \mu + (\xi_1 + \xi_2 - 1)\cdot\Delta, \quad \xi_1,\xi_2 \in U(0,1)$$



Test - procedura:

1. Generujemy serię N liczb z rozkładu
2. Grupujmey liczby w k przedziałach
3. Liczymy wartość statystyki testowej χ^2
4. Dla zadanego poziomu istotności α sprawdzamy czy
   χ^2 < wartości granicznej

$$\chi^2_{k-2-1} = 4.52 < 14.06\,(\alpha = 0.05)\text{- nie odrzucamy H0}$$