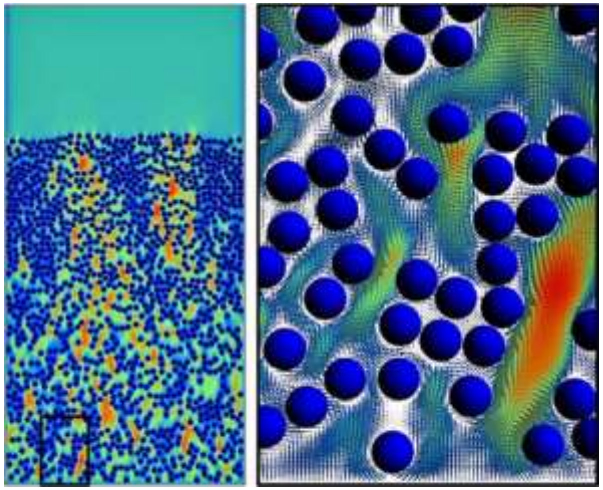
NÁHODNÁ ČÍSLA V NUMERICKÝCH VÝPOČTECH

Náhodná čísla

- Co jsou náhodná čísla:
  - postrádají jakýkoliv vzor (sekvenci, předvídatelnost)
- Proč potřebujeme náhodná čísla?
- Pro vytvoření náhodného stavu nebo náhodné události
  - kryptografie (generování klíčů)
  - simulace fyzikálních procesů
    - náhodné souřadnice molekul
    - Brownův pohyb, chaos
    - radioaktivní rozpad, šum, …
  - modelování lidského chování
    - pohyb lidí, rozhodovací procesy, …
  - videohry
    - nepředvídatelnost
    - události, pohyb nepřátel, směr výstřelu, …
- Nejčastější aplikace
  - počítačové simulace
  - videohry
  - …



Možnosti získání náhodných čísel

- **Generování náhodných čísel**
  - základem vždy pozorování reálného jevu
  - co použít při simulacích?

- Dvě možnosti
- Přímé **měření reálného jevu**
  - skutečná náhodnost
    - házení mincí, kostkou, měření přírodních jevů, …)
  - malý počet náhodných hodnot
  - náročný proces

- **Odhad pravděpodobnostního rozdělení** pozorovaného jevu
  - odhad rozdělení a jeho parametrů (na základě sběru dat)
  - **generování hodnot** z tohoto rozdělení
  - dostatečný počet „náhodných“ hodnot
  - pseudonáhodná čísla
    - John von Neumann, 1946 prvotní metoda pro rychlé získání náhodných číslic

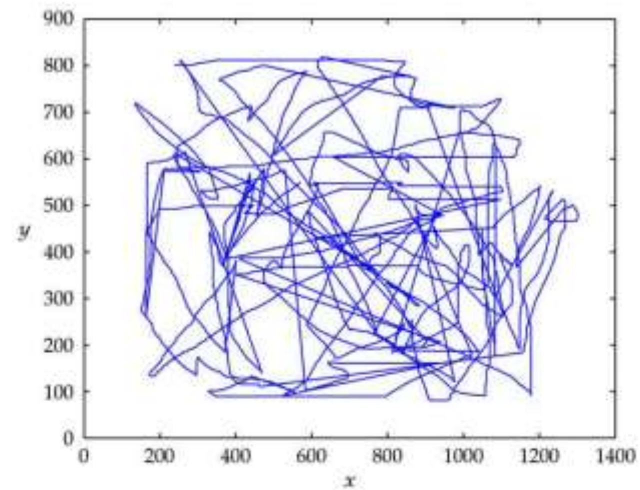
Generátory náhodných čísel

- Generátory náhodných čísel
  - fyzikální generátory
  - tabulky náhodných čísel
  - vypočtená pseudonáhodná čísla
    - lineární kongruentní generátory

- Nevýhody pseudonáhodných čísel
  - nemají zcela náhodné rozdělení
    - problém (šifrovací aplikace, hry, …)
  - zkoumáním minulé sekvence lze často určit následné číslo
    - např. neuronovou sítí
  - stále lepší metody , ale i rozvoj oblasti hlubokého učení
    - schopné rozpoznat následnou sekvenci čísel
- Základní generátory
  - Randu, ZXSPECTRUM, Marsaglia XorShift, Mersene Twister, Park&Miller,...

Skutečná náhodná čísla

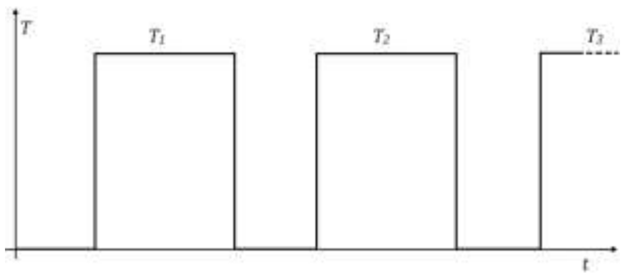
- Generátory skutečně náhodných čísel
  - poskytují skutečně náhodná čísla
  - využívají nějaký **fyzický** nebo **fyzikální** jev
    - může být zpřístupněno webovými službami



- Generátory založené na **fyzických** jevech
  - např. generování čísla ze sekvence pohybu myší
    - nebo úhozů do klávesnice (prodlevy)
  - problémy
    - člověk může využívat podobný vzor pohybů a úhozů
    - komunikace s OS pomocí bufferů, které mohou náhodnost vyrušit
- Generátory založené na **fyzikálních** jevech
  - lepší, ale finančně náročnější

Fyzikální generátory

- Nejčastěji využívané jevy
  - radioaktivní rozpad nuklidů
    - Geiger-Müllerův počítač
  - atmosférický šum
    - elektromagnetické vlnění v daném prostoru a čase
    - lze získat citlivou anténou
  - akustický tlak v místnosti (šum z hluku)
    - slabší generátory
    - problém s prediktivními jevy jako hluk z otáček větráku

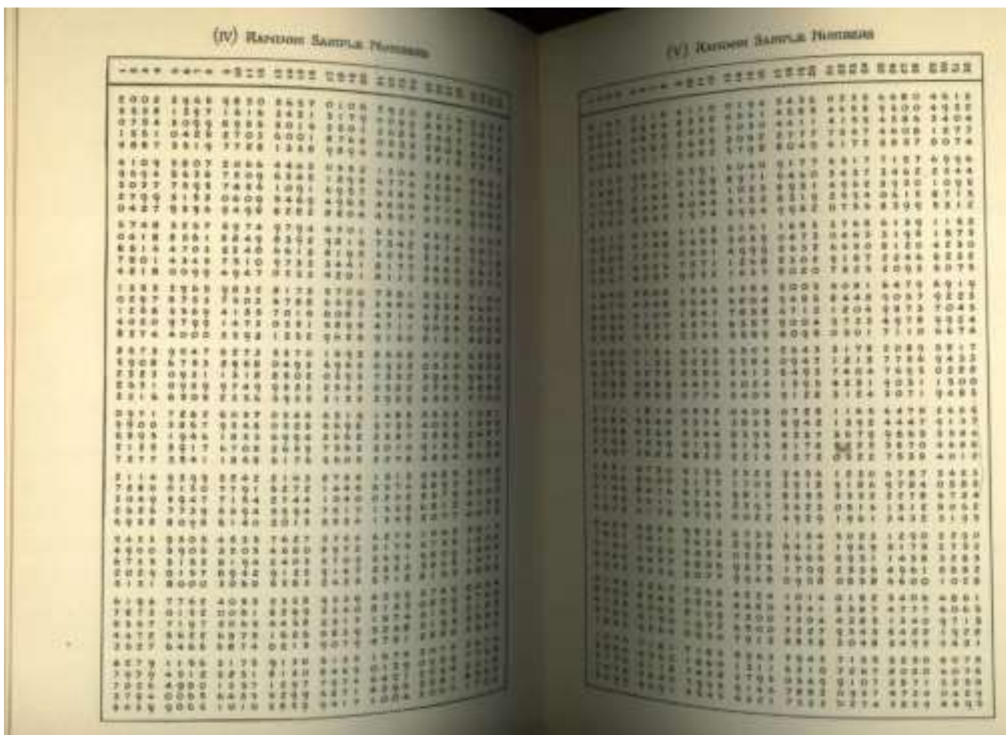


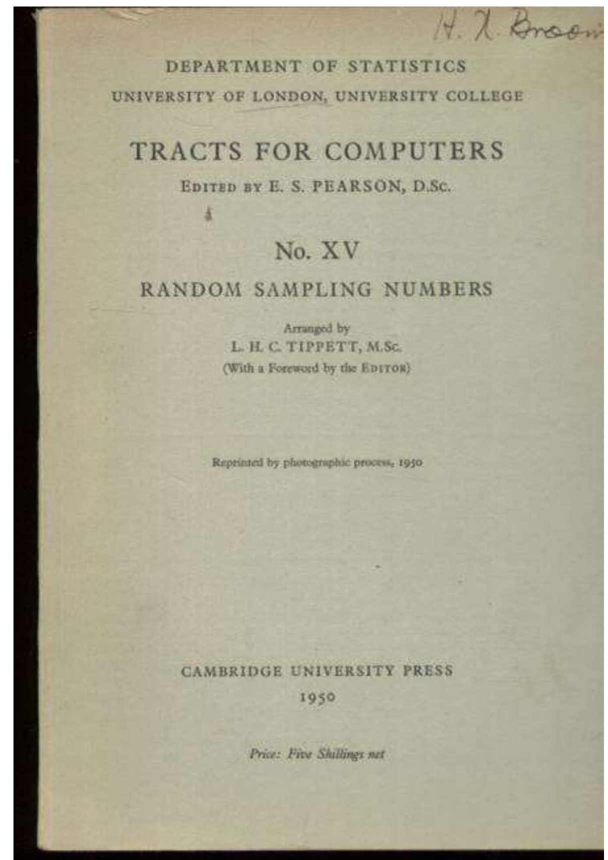
pokud $T_1 > T_2$ – zapíšeme 0
pokud $T_1 < T_2$ – zapíšeme 1

- Využití externích webových služeb přes jejich rozhraní REST
  - levné, spolehlivé
  - např. random.org
    - data pro vygenerování náhodných číslic z atmosférického šumu
    - potřebujete získat API klíč
    - uvádíte v požadavku ve formátu JSON pomocí metody POST z HTTP protokolu
    - klíč můžete vygenerovat po registraci na stránce: https://accounts.random.org/create
    - při volbě developer licence je registrace zdarma, ale denní limit vygenerovaných čísel 1000

Tabulky náhodných čísel

- Opravdu náhodná čísla
  - z fyzikálních generátorů
    - disk, CD nosič, páska
  - rozsáhlé soubory dat (tabulky)
    - 1927 – Tipper - 40 tis náhodných čísel
    - 1955 – RandCorp - 1 mil. náhodných čísel





H. L. Broom

DEPARTMENT OF STATISTICS
UNIVERSITY OF LONDON, UNIVERSITY COLLEGE

TRACTS FOR COMPUTERS

EDITED BY E. S. PEARSON, D.Sc.

No. XV

RANDOM SAMPLING NUMBERS

Arranged by
L. H. C. TIPPETT, M.Sc.
(With a Foreword by the EDITOR)

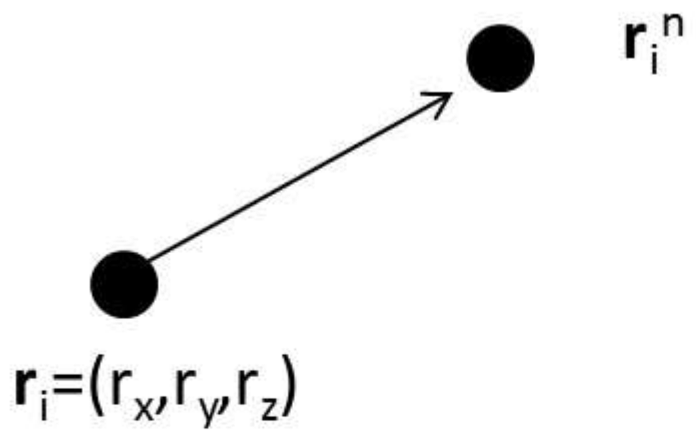
Reprinted by photographic process, 1950

CAMBRIDGE UNIVERSITY PRESS
1950

Price: Five Shillings net

Využití náhodných čísel

- Např. náhodný posun částice v kapalině
  - generování náhodného čísla ξ v intervalu $(0,1)$
  - transformace ξ podle odhadnutého rozdělení
    - které lépe popisuje chování částice (např. normální rozdělení)
  - realizace náhodného jevu (sledovaná veličina)
    - posunutí o náhodný vektor



- máme dané maximální posunutí $\boldsymbol{r}_{max}$
- generujeme náhodný směr

$$\underline{\xi} = (\xi_{x'}, \xi_{y'}, \xi_{z'})$$

- částici posuneme na nové (náhodné) místo

$$\boldsymbol{r}_i^n = \boldsymbol{r}_i + \underline{\xi} \cdot \boldsymbol{r}_{max}$$

Transformace náhodných čísel

- Transformace na jiné rozdělení/interval
  - náhodné číslo ξ je v intervalu (0,1), $\langle 0,1 \rangle$, …
  - nejčastější lineární transformace
  - $\zeta^T = \xi(b - a) + a, \qquad a, b \in \mathbb{R}$

- **Lze náhodný posun částice generovat i jiným způsobem?**
- Monte Carlo simulace
  - opakované generování náhodných posunů
    - a vyhodnocování jejich dopadu na chování
  - získáme distribuci možných poloh částice
    - a odhadneme pravděpodobnostní rozdělení
- deterministické algoritmy s náhodnými začátečními podmínkami
  - sledování vzájemných interakcí částic
    - stochastický charakter není zásadní

VYPOČÍTANÁ NÁHODNÁ ČÍSLA

Vypočítaná náhodná čísla

- Pseudonáhodná (kvazináhodná)
  - Algoritmus (posloupnost čísel), perioda P
  - náhodné číslo $\xi_{i+1} = f(\xi_i, \xi_{i-1}, \xi_{i-2}, \dots \xi_{i-n})$

- Von Neumannovy generátory
- Lineární kongruentní generátory
- Generátory s posuvnými registry

- Požadavky
  - co nejdelší perioda P
  - co největší náhodnost v rámci periody a rovnoměrné pokrytí
  - co největší rychlost generování čísla ξ_{i+1}

Vypočítaná náhodná čísla

- Von Neumannovy generátory (1946)
- „Každý, kdo se zabývá aritmetickými metodami vytváření náhodných čísel, se nepochybně dopouští hříchu“

- První známý generátor pseudonáhodných čísel
  - krátká perioda opakování náhodných čísel
  - řešil problém pomalého čtení náhodných čísel z děrných štítků pro počítač ENIAC

- Metoda prostředku čtverce (Middle-square)
    - (prostředních řádů druhé mocniny)
  - Zvolím počáteční číslo x_0 o $2k$ číslicích
  - Číslo se umocní
  - Z druhé mocniny se vybere prostředních 2k číslic
  - Získané číslo je dalším prvkem posloupnosti

Vypočítaná náhodná čísla

- Von Neumannovy generátory (1946)

x_0	x_0^2
1236	01527696

x_1	x_1^2
5276	27836176

x_2	x_2^2
8361	69906321

x_3	x_3^2
9063	81237969

- Krátká perioda P
- Malá náhodnost v rámci periody
- Pomalý proces generování

Lineární kongruentní generátory

- Historicky jeden z nejdůležitějších generátorů pseudonáhodných čísel
  - používán v mnoha implementacích novějších generátorů
    - např.: Park-Miller z C++11 standardní knihovny

- Princip:
  - 1. zvolíme parametr $\boldsymbol{M}$ (modulus)
    - prvočíslo nebo jeho mocninu
  - 2. zvolíme parametr $\boldsymbol{C}$ (inkrement)
    - pro $C = 0$ se nazývá generátor Lehmerův
  - 3. zvolíme parametr $\boldsymbol{a}$ (násobek)
  - 4. algoritmus vyžaduje semínko seed, které představuje první ξ_i
  - 5. další náhodné číslo ze vzorce: $\xi_{i+1} = (\boldsymbol{a} * \xi_i + \boldsymbol{C}) \bmod \boldsymbol{M}$

Lineární kongruentní generátory

- D. H. Lehmer (1948)

$$\xi_{i+1} = (a_0\xi_i + a_1\xi_{i-1} + \cdots + a_n\xi_{i-k} + b)(\mathrm{mod}\ M)$$

- Konstanty: a_j, b, M
  - vhodnou volbou konstant určujeme vlastnosti generátoru
  - $k > 0, \xi_i < M$
- Semínko (násada)
  - $\xi_0, \dots, \xi_{-k}$ $(i = 0)$
  - $\xi_1 = (a_0\xi_0 + a_1\xi_{-1} + \cdots + a_n\xi_{-k} + b)(\mathrm{mod}\ M)$
- **Stejná násada = stejná posloupnost čísel**

$$\xi_i = 0, 1, 2, \dots, M-1, \qquad \xi_i = \frac{\xi_i}{M} \rightarrow \qquad \xi_i \in (0,1)$$

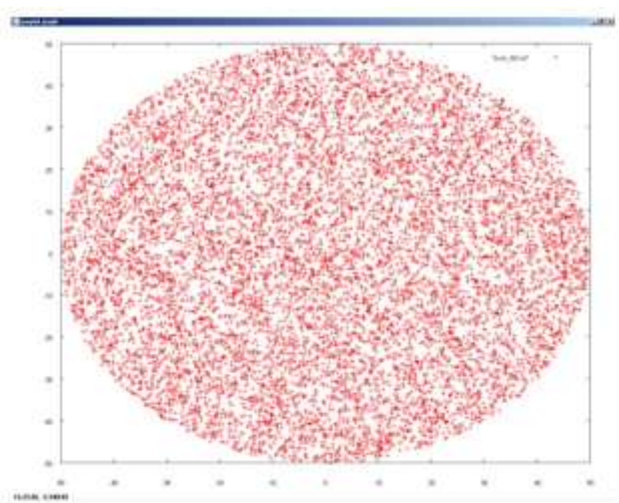
- Dělení
  - Multiplikativní generátory
  - Aditivní generátory
  - Smíšené generátory

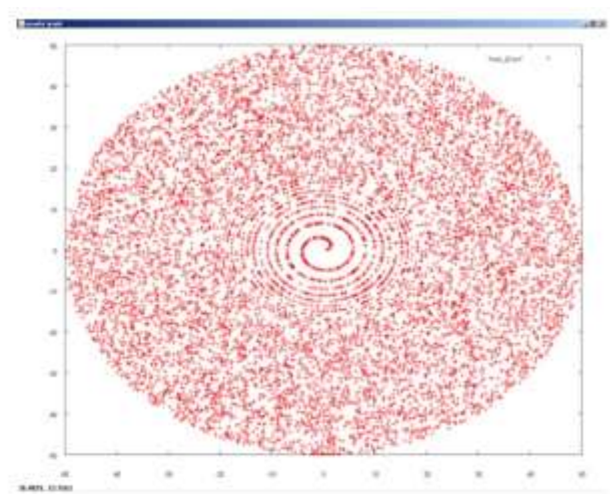
Multiplikativní LCG

- Multiplikativní generátory
  - velmi rychlé generování

D. H. Lehmer $\xi_{i+1} = a_0 \xi_i \pmod{M}$

- IBM 360: $\xi_{i+1} = 7^5 \xi_i \pmod{2^{31} - 1}$
- IBM 370: $\xi_{i+1} = 630360016\, \xi_i \pmod{2^{31} - 1}$
- Fortran: $\xi_{i+1} = 13^{13} \xi_i \pmod{2^{59}}$
- ZX SPECTRUM: $\xi_{i+1} = 75 \xi_i \pmod{65537}$
- RANDU: $\xi_{i+1} = a_0 \xi_i \pmod{2^{31}},\ a_0 = 1$
- jazyk BASIC (nevhodná volba konstant)





Aditivní LCG

- Fibonacciho generátor
  - využívá Fibonacciho posloupnost
    - předposlední a poslední hodnotu
  - $\xi_{i+1} = (\xi_i + \xi_{i-1}) (\mathrm{mod}\ M)$
- Opožděný Fibonacciho generátor (Mitchell, Moore, 1958)
  - využívá obecné hodnoty opožděnosti
    - nepoužívá se, ale jednoduchý
  - $\xi_{i+1} = \left(\xi_{i-j} + \xi_{i-k}\right) (\mathrm{mod}\ M)$
  - místo "+" může být jiná operace
    - $+ - * /$ ap.
- Millerův-Prenticův generátor
  - $\xi_{i+1} = (\xi_{i-1} + \xi_{i-2n}) (\mathrm{mod}\ 3137)$ $(P = 9.8 \cdot 10^6)$
- Další aditivní generátory
  - $\xi_{i+1} = (\xi_{i-5} + \xi_{i-17}) (\mathrm{mod}\ M)$

Volba M ovlivňuje periodu generátoru

$M = 2^6 \rightarrow P = 1.6 \cdot 10^7$

$M = 2^{16} \rightarrow P = 4.3 \cdot 10^9$

$M = 2^{32} \rightarrow P = 2.8 \cdot 10^{14}$

Smíšené LCG

- Smíšené generátory
  - kombinují vlastnosti multiplikativního a aditivního
    - tak, aby došlo ke zlepšení vlastností
  - $\xi_{i+1} = (a\xi_i + b)(\mathrm{mod}\ M)$
- Příklady
  - $\xi_{i+1} = (69069\xi_i + 1)(\mathrm{mod}\ 2^{32})$
    - dlouhá perioda
    - simulace, kryptografické funkce

Minimal Standard

- Lewsi, Goodman, Miller (1969)
  - $\xi_{i+1} = a_0 \xi_i (\mathrm{mod}\ M)$, $\quad a_0 = 16\,807$, $\quad M = 2^{31} - 1 = 2\,147\,483\,647$
  - generátor má plnou periodu (rovnoměrné pokrytí)
    - vybraná subsekvence čísel je nerozeznatelná od celé, generované sekvence
  - generátor projde **testy spolehlivosti**
  - lze jej **efektivně** implementovat pomocí 32-bitové architektury
    - dnes zastaralý, ale stále se používá pro jednoduchost
  - snadná implementace do vysokoúrovňových jazyků
  - seed generátor: $\xi_0 \in (1,\ 2^{31} - 1)$, jsou rovnocenné

Generátory pseudonáhodných bitů

- Posuvné registry
  - struktury k ukládání a posouvání jednotlivých bitů
    - o jeden nebo více míst
    - vpravo nebo vlevo (shift right, shift left)
    - realizace pomocí kombinační logiky nebo speciálních posuvných instrukcí procesoru
  - jeden nebo více registrů
    - každý uchovává určitý počet bitů
- pro výpočet následujícího čísla se používá kombinace bitů z několika registrů
  - některé bity se mohou použít jako zpětná vazba do registru
- počáteční stav registrů = semínko
  - ovlivňuje celou posloupnost čísel
  - pokud se použije stejné, bude posloupnost stejná
- Vlastnosti
  - konečná perioda
  - jednoduchost
  - malá paměťová náročnost
  - rychlost

Pravidlo 30

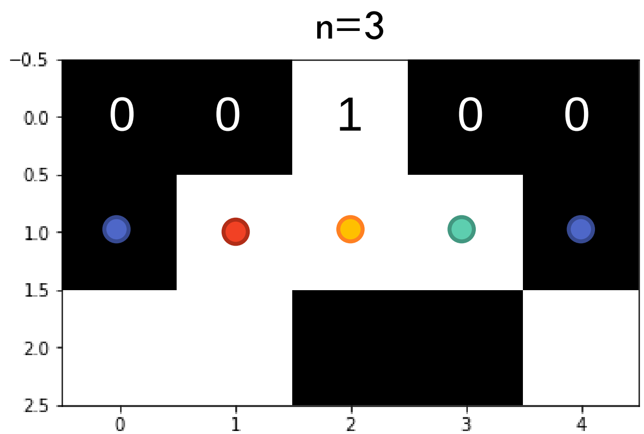
- Stephen Wolfram (1983)
  - název podle decimální reprezentace schémata binárních operací
- Pravidlo 30
  - jedno z nejzajímavějších výpočetních schémat základních automatů
    - pro aktualizaci stavu buněk v 1D buněčném automatu
  - vytváří automat, obsahující pseudonáhodné sekvence
    - aperiodické chaotické sekvence bitů
- Vlastnosti
  - jednoduchá definice
  - přesto velmi komplexní a chaotické vzorce
    - použijeme jako náhodnou posloupnost
- Princip
  - výpočet stavu každé buňky v novém řádku
    - na základě stavů 3 sousedících buněk v předchozí řadě
    - podle pravidla $a \text{ xor } (b \text{ or } c)$

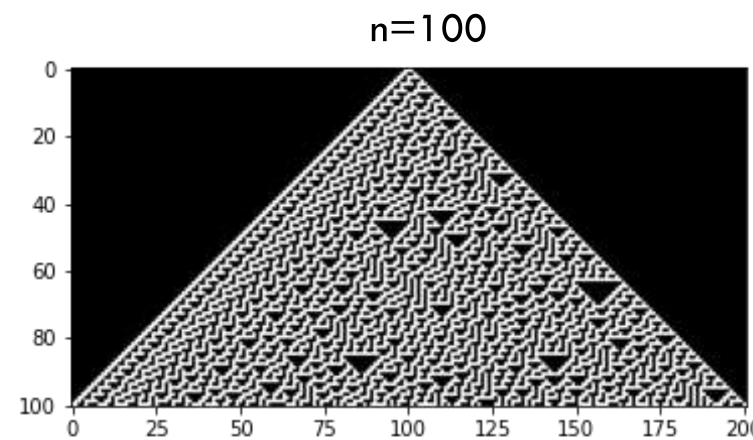
Pravidlo 30

- Stav každé buňky v novém řádku
  - na základě stavů tří sousedících buněk v předchozí řadě
    - podle pravidla $a \text{ xor } (b \text{ or } c)$

$a \text{ xor } (b \text{ or } c)$

$111 \rightarrow 0$
$110 \rightarrow 0$
$101 \rightarrow 0$
$100 \rightarrow 1$
$011 \rightarrow 1$
$010 \rightarrow 1$
$001 \rightarrow 1$
$000 \rightarrow 0$





- Postup
  - 1. Nastav prvotní řádek na binární 0, prostřední bit na 1
  - 2. Proveď vývoj automatu do zvolené generace pomocí pravidla 30
  - 3. Vyber prostřední sloupec, přeskoč N bitů (semínko) a získej 8 binárních číslic
  - 4. Vytvoř z M binárních číslic pseudonáhodné číslo

Generátory s posuvnými registry

- S posuvnými registry s lineární zpětnou vazbou
  - posuvný registr délky L
    - L = $\{R_0, R_1, \ldots, R_{L-1}\}$ vnitřních registrů
    - časový signál
    - charakteristický mnohočlen C(x)
- Princip
  - v každém časovém okamžiku se:
    - obsah R_i přesune do R_{i-1}, R_0 se předá na výstup
    - do registru R_{L-1} se uloží (výpočtem) nový obsah
  - R_i uchovává jednu jednotku informace
    - Jeden bit (0,1), 2^n bitů (velikost slova / jednotka času)
  - vnitřní stav generátoru S = $\{S_0, \ldots, S_{L-1}\}$
  - charakteristický mnohočlen

$$C(x) = 1 + c_1 x + c_2 x^2 + \cdots + c_L x^L$$

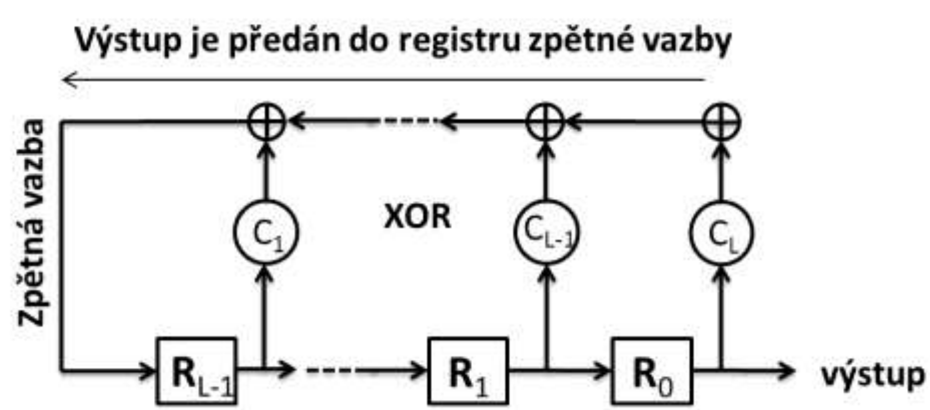
  - koeficienty: zbytek po dělení modulo
    - 1 – liché, 0 – sudé

Iniciace generátoru

Pozor na zakázané stavy logické funkce

Zacyklení generátoru

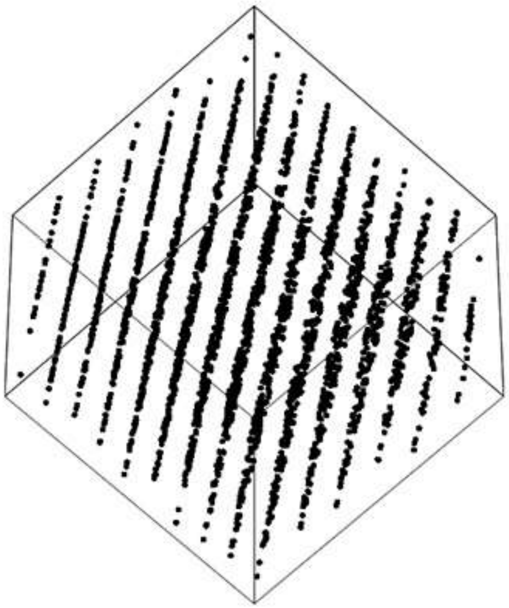
Iniciace pomocí LCG (Lehmer,…)



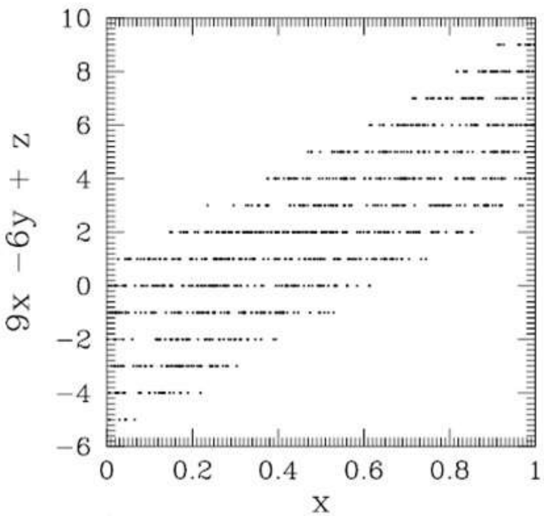
Marsaglia XORshift

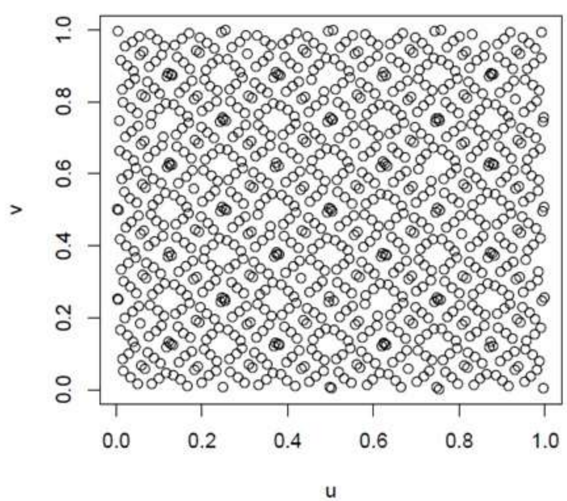
- George Marsaglia (2003)
    - americký matematik (statistik)
  - na základě von Neumannova generátoru
- Užití operace XOR a bitového posunu
  - XOR (exkluzivní disjunkce)
    - logická operace, výstupem pravda, pokud vstupy unikátní
    - $A = (0,1,0,1), \quad B = (0,0,1,1), \quad A \oplus B = (0,1,1,0)$
  - logické shift (bitový posun)
    - levý posun: 001010111 → 01010111**0**
    - pravý posun: 001010111 → **0**00101011
- Postup
  - bitové posuny (různé podle implementace)
    - např. Xorshift128 o 23, 17, 26 a 11 míst vlevo
  - posunuté bity se použijí pro XOR s jinými bity v registru
    - aby se vytvořil nový stav generátoru
    - aplikujeme XOR na vektor β (binární) s posunutou verzí sebe sama
    - $\beta >> a$ – bitový posuv doprava o a pozic
      - a je parametr generátoru
    - $\beta \oplus (\beta >> a)$ – XORshift vektoru β o a pozic doprava

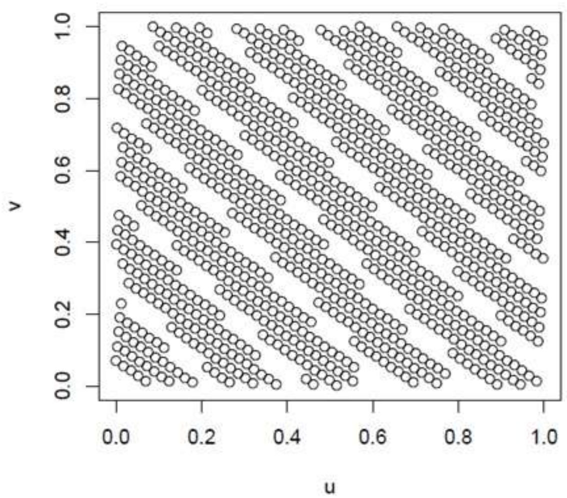
Marsaglia XORshift



- Marsaglia XORshift produkuje sekvenci:
  - $2^{32}-1$ x celých čísel
  - $2^{64}-1$ x, y dvojic
  - $2^{96}-1$ x, y, z trojic
- Nevýhody
  - „Random numbers fall mainly in the planes.“
    - náhodná čísla padají častěji do nadrovin multidimenzionálního prostoru
- Další Marsagliovy algoritmy: Ziggurat, …
  - ke generování náhodných čísel s normálním rozdělením
  - multiplikativní generátory: „Crystalline“ nature







Mersenne Twister

- Makoto Mastumoto, Takuji Nishimura, 1998
  - Mersenneho prvočíslo
    - prvočíslo o jedničku menší než mocnina 2, $M = 2^n - 1$
    - $2^3 - 1 = 7$ (je prvočíslo), $M = 2^4 - 1 = 15$ (není prvočíslo)
  - www.mersenne.org
    - největší ověřené Mersenneho prvočíslo (r. 2021) 57 885 161 (48. Mersenneho prvočíslo)
    - r. 2006 32 582 657 (44.)

- Použití
  - dnes jeden z nejpoužívanějších
  - různé verze MT
    - nejvyužívanější verze (nastavení parametrů) MT19937
  - používá Python v modulu Random
  - dále R, Ruby, Free Pascal, PHP, Maple, MATLAB, GAUSS, Julia, Microsoft Visual C++,...
    - numpy jiný (PCG64 z roku 2014)

Mersenne Twister

- Založen na generátoru s posuvnými registry
  - polynom $C(x)$ s řádem P, stavový vektor x o velikosti w bitů

- Generuje stavový vektor x

$$x_{k+n} = x_{k+m} \oplus \left(x_k^u \mid x_{k+1}^l\right)A$$

  - x_n je stavový vektor v kroku n
  - x_k^u je subvektor složený z w-r levých bitů vektoru **x**
  - x_{k+1}^l je subvektor složený z w-r pravých bitů vektoru **x**
  - | představuje operaci zřetězení
    - "Hello" | "world" → "hello world"
  - A je transformační matice
  - $\oplus$ XOR

- Vlastnosti
  - velice dlouhá perioda
    - až $P = 2^{19937} - 1$
  - TinyMT (2012)
    - $P = 2^{512}$, ale méně zatěžuje procesor

<u>Špatný Seed:</u>
trvá dlouho, než začne generovat náhodnou sekvenci

Testy generátorů náhodných čísel

- Mějme uspořádanou k-tici náhodných celých čísel $n(k)$
- Testy náhodnosti
  - Statistické testy (chí kvadrát test)
  - Transformace (Hadamard, Marsaglia)
  - Komplexita (složitost)
    - Kolmogorovova komplexita: měří složitost k-tice podle počtu znaků, délky programu, který takovou k-tici vyprodukuje
- DIEHARD testy
    - George Marsaglia 1995, Stanfordova univerzita
  - sada statistických testů a algoritmů
  - prověřují různé aspekty náhodnosti generovaných číselných posloupností
    - statistická podobnost skutečným náhodným posloupnostem
    - požadované vlastnosti náhodnosti (rovnoměrné rozložení, minimální korelace mezi čísly)
- TESTU01
  - softwarový balík
  - vylepšená verze DIEHARD

Testy generátorů náhodných čísel

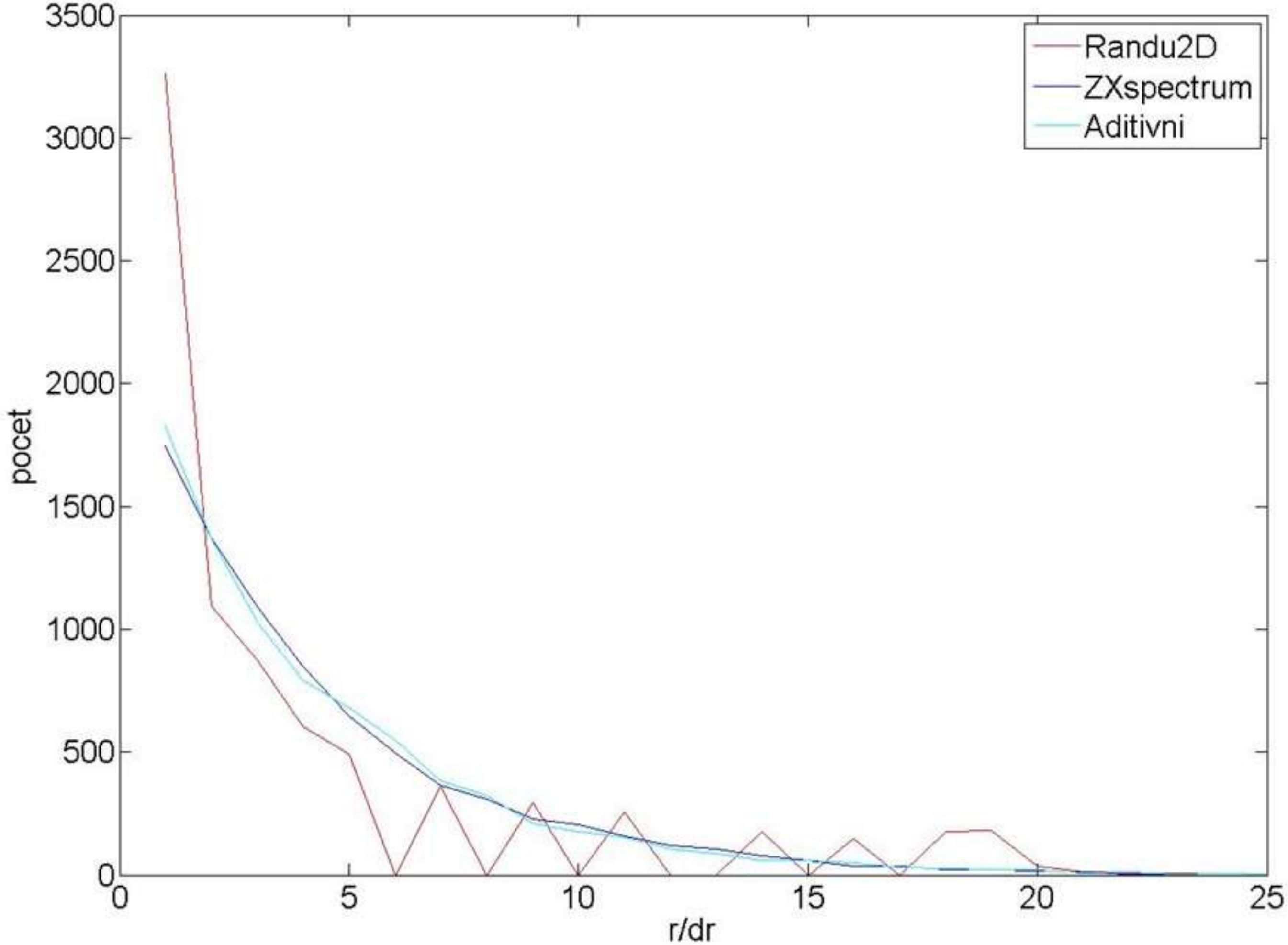
- DIEHARD testy

1. Testy na rovnoměrné rozložení
   - zda je rozložení čísel rovnoměrné v daném intervalu
2. Testy na korelaci
   - mezi jednotlivými čísly v posloupnosti
3. Testy na různorodost
   - různorodost hodnot v posloupnosti
   - další vlastnosti, např. počet unikátních hodnot
4. Testy na sériovou korelaci
   - detekce opakujících se vzorů
5. Testy na permutace
   - kontrola vlastností permutací čísel

Testy generátorů náhodných čísel

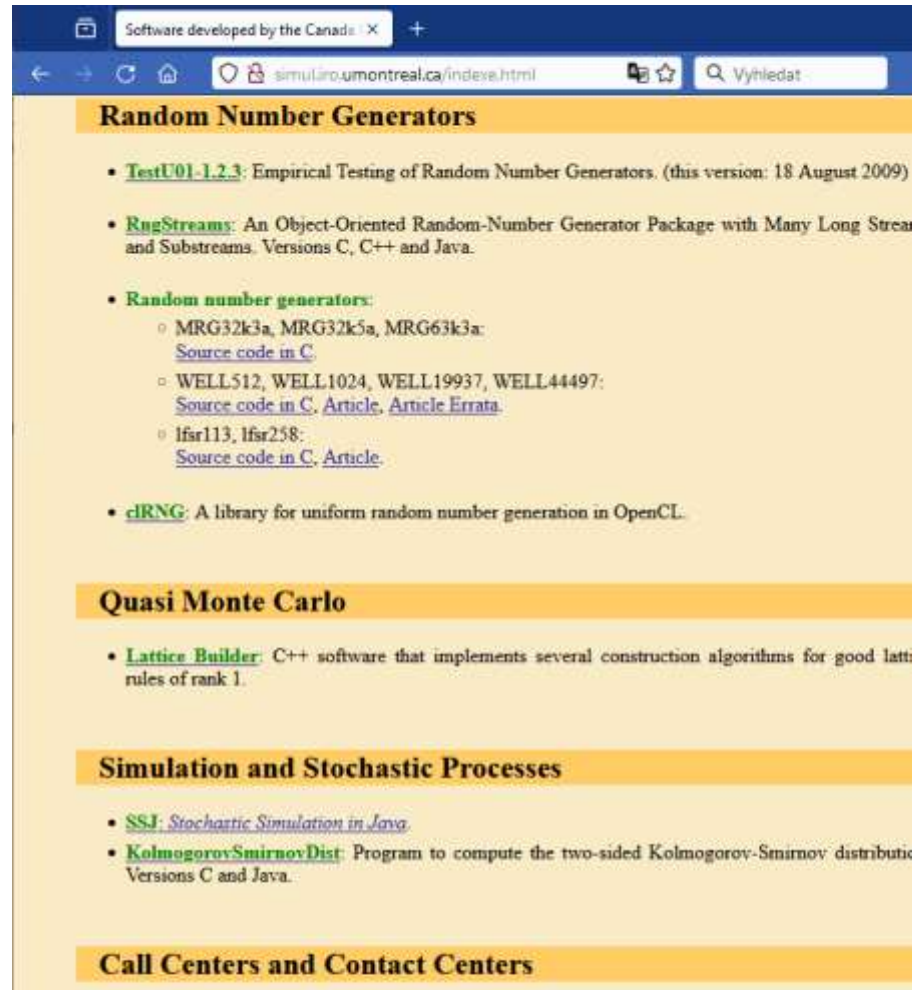
- DIEHARD testy
  - Birthday spacings
    - Vyberte náhodně 2 body na generovaném intervalu, vzdálenost mezi nimi podléhá Poissonovu rozdělení
  - Minimum distance test
    - Náhodně umístěte n bodů, změřte vzdálenosti d mezi všemi $\frac{n(n-1)}{2}$ páry. Hledáme minimální vzdálenost.
      - veličina d^2 bude podléhat exponenciálnímu rozdělení se střední hodnotou 0.995
  - Runs test
    - Generujte náhodná čísla na intervalu (0,1) a počítejte, kdy dojde k růstu a kdy k poklesu
      - tyto počty splňují určité rozdělení
  - Parking lot test
    - Náhodně umístěte jednotkovou kružnici do čtverce o hraně 100 bodů; pokud se kružnice překrývají, zkuste to znovu; po 12,000 pokusech by počet úspěšně umístěných kružnic měl splňovat normální rozdělení
  - Count-the-1's test on a stream of bytes
    - Uvažuje řetězec bajtů. Každý bajt může obsahovat 0 až 8 jedniček
      - s pravděpodobností 1/256, 8/256, 28/256, 56/256, 70/256, 56/256, 28/256, 8/256, 1/256.
    - Nyní nechme vytvořit řetězec překrývajících se 5-písmenných slov.
    - Každé písmeno nabývá hodnot A, B, C, D, a E. Písmena jsou určována počtem jedniček v bajtu:
      - 0, 1 nebo 2 znamená A, 3 B, 4 C, 5 D a 6, 7 nebo 8 E.
    - Existuje 55 možných 5-písmenných slov, jejich výskyty se spočítají z řetězce 256 000 5-písmenných slov.

Minimum distance test



Testy generátorů náhodných čísel

- TestU01 (L'Ecuyer, Simard, Université de Montréal 2007)
  - vylepšená verze DIEHARD testů
    - rovnoměrnost, nezávislost, uniformita, rozptyl, …
  - knihovna v ANSI C
  - několik modulů
    - implementace generátorů (předprogramované)
    - implementace statistických testů
    - implementace známých sad testů
    - aplikace testů na generátor
    - SmallCrush, Crush, BigCrush: sady testů
  - Implementované generátory:
    - http://simul.iro.umontreal.ca/indexe.html



GENEROVÁNÍ JINÝCH ROZDĚLENÍ

Rozehrání náhodné veličiny

- Generování a transformace náhodné veličiny (rozehrání NV)

- Co umíme
  - generace náhodných čísel z rovnoměrného rozdělení
- Co potřebujeme
  - náhodnou veličinu X s jiným typem rozdělení
  - se zadanou hustotou pravděpodobnosti $f_X(x)$ či distribuční funkcí $F_X(x)$

- Metody
  - pro diskrétní NV
  - pro spojité NV
    - metoda inverzní funkce
    - metoda výběru
    - metoda superpozice

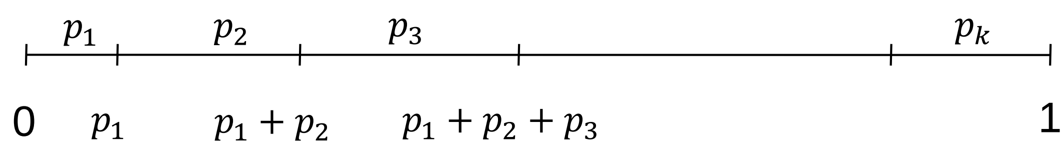
Rozehrání diskrétní náhodné veličiny

- Náhodná veličina (NV) X

$$X = \begin{pmatrix} x_1 & x_2 & \dots & x_n \\ p_1 & p_2 & \dots & p_n \end{pmatrix} \qquad p_i = P(X = x_i)$$

- Vytvoření vektoru (o n složkách)

$$(p_1, \quad p_1 + p_2, \quad p_1 + p_2 + p_3, \quad \dots, \quad 1)$$



- Vygenerujeme číslo y z rovnoměrného rozdělení $R(0,1)$
- Určíme, do kterého intervalu padne interval a jemu odpovídající NV X
  - podle podmínky

$$y < \sum_{i=1}^{j} p_i$$

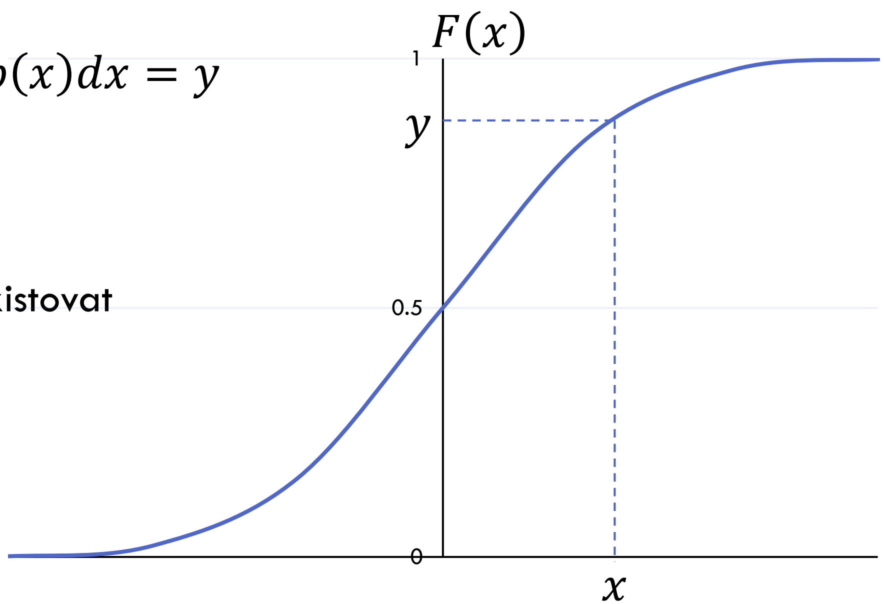
  - První interval j, pro který bude tato podmínka splněna, určí příslušnou hodnotu $X = x_j$

Metoda inverzní funkce

- Rozehrání spojité náhodné veličiny
  - hledáme NV X, hustota pravděpodobnosti $p(x)$, distr. funkce $F(x)$
  - máme NV Y s rovnoměrným rozdělením $R(0,1)$
    - vygenerujeme náhodné číslo $y \epsilon \langle 0,1 \rangle$
  - potom náhodná veličina $X = F^{-1}(Y)$ má rozdělení s distribuční funkcí $F(x)$

$$y = F(x) \quad \Longrightarrow \quad x = F^{-1}(y)$$

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} p(x)dx = y$$



řešíme tuto rovnici, analytické řešení nemusí existovat
výsledkem transformační vztah $x = g(y)$

Vyžaduje znalost analytické formy distribuce

Metoda inverzní funkce – příklady

- Mějme spojitou náhodnou veličinu X s kumulativní distribuční funkcí

  $F(x) = 1 - \exp(-\sqrt{x})$ pro $x \geq 0$ (a rovno 0 jinak)

  - najdeme inverzní funkci $F(x)$ vyjádřením x z rovnice $F(x) = y$

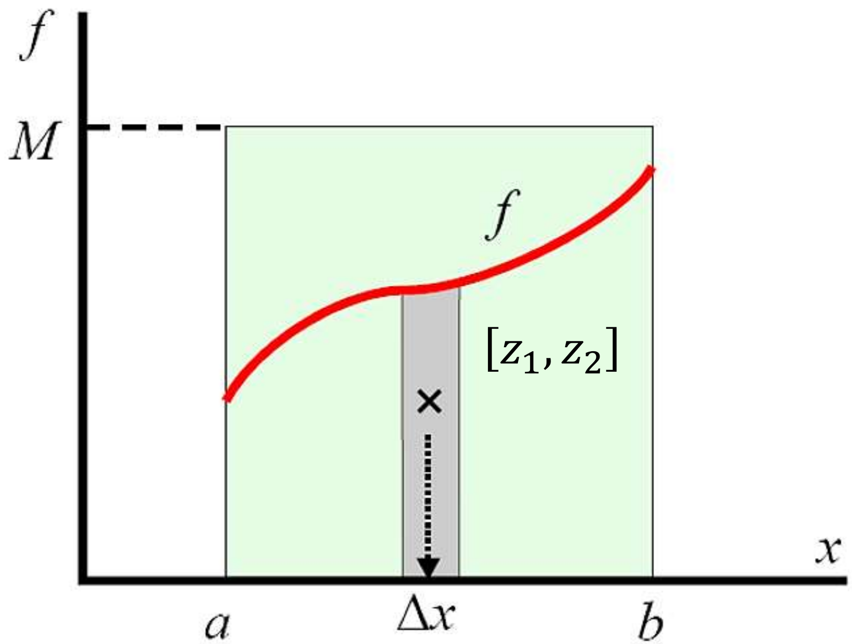
  $$x = (\log(1-y))^2$$

  - vygenerujeme náhodné číslo $y \in \langle 0, 1 \rangle$
  - transformujeme ho pomocí inverzní funkce
  - dostaneme náhodné číslo z rozdělení X

- Další příklady
  - $F(x) = \frac{x-a}{b-a} \quad \Longrightarrow \quad x = y(b-a) + a$
  - $F(x) = 1 - e^{-\lambda x} \quad \Longrightarrow \quad x = -\frac{1}{\lambda} \ln y$

Metoda výběru (von Neumannova)

- Metoda výběru (von Neumannova)
  - vhodná, když nelze analyticky vyjádřit inverzní funkci
  - $p(x)$ veličiny X je omezená na intervalu $\langle a, b\rangle$
  - volíme $M \geq \sup\big(p(x)\big)$
  - generujeme dvě náhodná čísla veličiny Y: y_1, y_2 (rovnoměrné rozdělení 0 až 1)
  - $z_1 = a + y_1(b-a)$ (x, mezi a a b)
  - $z_2 = My_2$ (y, pod M)
  - pokud bod (z_1, z_2) leží pod křivkou $p(x)$, volíme $x = z_1$, jinak opakujeme



Metoda superpozice (kompoziční)

- Metoda superpozice (kompoziční)
  - NV X s distrib. funkcí $F(x)$ je lineární kombinací jiných spojitých náhodných veličin
  - rozložíme její distribuční funkci $F(x)$ do tvaru

$$F(x) = \sum_{k=1}^{m} p_k F_k(x)$$

    - $F_k(x)$ jsou distribuční funkce, p_k pravděpodobnosti
    - $p_k > 0, \qquad p_1 + \cdots + p_m = 1$
  - Zavedeme diskrétní NV Z s rozdělením

$$Z = \begin{pmatrix} 1 & 2 & \dots & m \\ p_1 & p_2 & \dots & p_m \end{pmatrix} \qquad Z(z = k) = p_k \quad k = 1, \dots, m$$

  - generujeme 2 nezávislé hodnoty y_1 a y_2 veličiny Y (rovnom. rozdělení 0 až 1)
  - rozehrajeme číslem y_1 hodnotu $Z = k$; $\quad k = 1, \dots, m$ (číslo intervalu)
    - rozehrání diskrétní NV
  - z rovnice $F_k(x) = y_2$ určíme x
    - distribuční funkce veličiny X je rovna $F(x)$

Metoda superpozice – příklad

- Distribuční funkce $F(x)$ má tvar

$$F(x) = \sum_{k=1}^{m} p_k F_k(x)$$

  - řešíme rovnice

$$p_k > 0, \quad p_1 + \dots + p_m = 1$$

$$Z = \begin{pmatrix} 1 & 2 & \dots & m \\ p_1 & p_2 & \dots & p_m \end{pmatrix}$$

- Máme distribuční funkci odpovídající hustotě

$$p(x) = \frac{5}{12}\left(1 + (x-1)^4\right) \qquad x \in \langle 0, 2 \rangle$$

- rozložíme ji do tvaru

$$p(x) = \frac{5}{6} \cdot \left(\frac{1}{2}\right) + \frac{1}{6} \cdot \left(\frac{5}{2}(x-1)^4\right)$$

$$p_1(x) = \frac{1}{2} \qquad p_2(x) = \frac{5}{2}(x-1)^4$$

$$F_1(x) = \frac{x}{2} \qquad F_2(x) = \frac{1}{2}(x-1)^5$$

$$Z = \begin{pmatrix} 1 & 2 \\ \frac{5}{6} & \frac{1}{6} \end{pmatrix}$$

$$x = \begin{cases} 2y_1 & \text{pro } y_2 < \frac{5}{6} \\ 1 + \sqrt[5]{2y_1 - 1} & \text{pro } y_2 \geq \frac{5}{6} \end{cases}$$

Využití centrální limitní věty

- Nechť $X_1, \dots, X_n$ jsou náhodné veličiny z $R(0,1)$
- potom pro střední hodnotu a rozptyl jejich součtu platí

$$E\left(\sum_{i=1}^{n} X_i\right) = \frac{1}{2}n$$

$$D\left(\sum_{i=1}^{n} X_i\right) = \frac{1}{12}n$$

- k rozdělení $N(0,1)$ se pro $n \to \infty$ blíží veličina

$$X_n = \sqrt{\frac{12}{n}}\left(\sum_{i=1}^{n} X_i - \frac{1}{2}n\right)$$

- prakticky použitelný je vztah pro $n = 12$

$$X_{12} = \sum_{i=1}^{12} X_i - \frac{1}{2}n$$

- Pozn.: platí pro jakékoliv nezávislé a stejně rozložené náhodné veličiny X_i